Raytheon

Integrated Sensing and Processing (ISP) Phase II: Demonstration and Evaluation for Distributed Sensor Networks and Missile Seeker Systems

Progress Report:

1st Quarter Progress Report
1 March 2006 – 31 May 2006

Acknowledgment of Support
This material is based upon work supported by the United States Air Force under Contract No. N00014-04-C-0437.

Contract No.: N00014-04-C-0437
Contract Line Item Number 0001
Deliverable Item: Publications (A001-005)

Raytheon Company
P.O. Box 11337
Tucson, AZ 85734-1337

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
17 MAY 2000		

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
1st Quarterly Progress Report	**N00014-04-C-0437**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Harry Schmitt	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Raytheon Missile System,1151 Herman Rd ,Tucson,AZ,85706	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The primary goal of this effort is to bring to maturity a select set of basic algorithms, hardware, and approaches developed under the Integrated Sensing and Processing (ISP) Phase I program, implement them on representative hardware, and demonstrate their performance in a realistic field environment. We have identified a few promising research thrusts investigated in ISP Phase I where field demonstrations are cost prohibitive but collected data sets are available. Here, we will conduct a thorough performance evaluation.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**48**	

01 June 2006

Progress Report

CDRL A001 No. 5
Fifth Quarterly Progress Report for Period of Performance
1 March 2006 – 31 May 2006

Integrated Sensing Processor Phase 2
Program Manager: Dr. Harry A. Schmitt
Principal Investigator: Dr. Harry A. Schmitt

Sponsored By:

Defense Advanced Research Projects Agency/DSO
Dr. Carey Schwartz/DARPA DSO
Program Manager: Dr. Dan Purdy/ONR
Issued by ONR under Contract #N00014-04-C-0437

Prepared By:

Raytheon Systems Company
P.O. Box 11337
Tucson, AZ 85734

EXECUTIVE SUMMARY

The primary goal of this effort is to bring to maturity a select set of basic algorithms, hardware, and approaches developed under the Integrated Sensing and Processing (ISP) Phase I program, implement them on representative hardware, and demonstrate their performance in a realistic field environment. We have identified a few promising research thrusts investigated in ISP Phase I where field demonstrations are cost prohibitive but collected data sets are available. Here, we will conduct a thorough performance evaluation.

TABLE OF CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

0. Technical Abstract

Advances in sensor technologies, computation devices, and algorithms have created enormous opportunities for significant performance improvements on the modern battlefield. Unfortunately, as information requirements grow, conventional network processing techniques require ever-increasing bandwidth between sensors and processors, as well as potentially exponentially complex methods for extracting information from the data. To raise the quality of data and classification results, minimize computation, power consumption, and cost, future systems will require that the sensing and computation be jointly engineered. ISP is a philosophy/methodology that eliminates the traditional separation between physical and algorithmic design. By leveraging our experience with numerous sensing modalities, processing techniques, and data reduction networks, we will develop ISP into an extensible and widely applicable paradigm. The improvements we intend to demonstrate here are applicable in a general sense; however, this program will focus on distributed sensor networks and missile seeker systems.

1.0. Management Overview and Summary

1. A. Program Summary

The Raytheon Company, Missile Systems (Raytheon) ISP Phase II program is a twenty-four month contract with a Period of Performance (PoP) covering 1 March 2005 to 28 February 2007. Raytheon has four universities and one small business as ISP Phase II subcontractors: Arizona State University (ASU); Fast Mathematical Algorithms and Hardware (FMAH); Georgia Institute of Technology (Georgia Tech); Melbourne University (UniMelb) and the University of Michigan (UM).

1. B. Program Status

The Raytheon ISP Phase II Program status can be summarized as remaining "on track." All of the negotiations have been completed and all of the subcontractors are now under subcontract. We had incurred some schedule slips on both the distributed tracking and the Cooperative Analog Digital Signal Processing (CADSP) demonstrations during the previous PoPs. An updated schedule for the distributed tracking demonstration was developed during the previous PoP and is again included in Section 2.0. While the revised schedule still supports a demonstration before 28 February, there is little room for further slippage. The current status of the CADSP imager hardware is discussed in Section 2.A.6. Progress in the current PoP has been such that we are pretty much on schedule for both the distributed tracking and CADSP demonstrations

The Program is still running below the spending plan; however, we expect to complete the contract on time and budget. As of 28 April 2006, 38% of contract funds had been expended with ~51% of the program complete. In part, the contract expenditure reflects an under-run due to delays in receiving invoices from our subcontractors. We are still working this issue. Raytheon has improved its spending profile significantly. While we are still under-run on an inception-to-date basis (41% of funds expended), our staffing level of ~3.5 heads has us completing the contract just slightly under budget. Initially the reduced Raytheon spending was designed to better align with the subcontractor schedules; however, we have also encountered some difficulties with personnel availability. The problem of Raytheon personnel availability remains a concern, but it has largely been resolved.

One area of significant concern is the availability of a suitable radar test and integration engineer. We continue to work this issue, but feel that we have an acceptable solution. A junior test engineer has been assigned to support our program, and we have gotten a commitment from a senior radar analyst for consulting. This problem remains one of the higher risks for our program.

1. C. Personnel Associated/Supported

Raytheon

Dr. Harry A. Schmitt	Principal Investigator
Mr. Donald E. Waagen	Co-Principal Investigator
Dr. Sal Bellofiore	Distributed Sensing Lead
Mr. Thomas Stevens	Distributed Sensing Support
Dr. Robert Cramer	Mathematical Support
Mr. Craig Savage	Waveform Design and Control Lead
Dr. Nitesh Shah	High Dimensional Processing Data Lead
Mr. William Daniels	Radar Test and Integration Support

FMAH

Professor Paolo Barbano
Professor Ronald Coifman
Dr. Nicholas Coult

ASU

Professor Darryl Morrell
Professor Antonia Papandreou-Suppappola

Georgia Tech

Professor David Anderson
Professor Paul Hasler

UniMelb

Dr. Barbara LaScala
Professor William Moran
Dr. Darko Musicki
Dr. Sofia Suvorova

UM

Professor Al Hero
Dr. Neal Patwari

Significant Personnel Actions: There was one significant personnel change during the current PoP. Mr. William Daniels was added for radar test and integration support.

1. D. Recent Accomplishments and Events

An amended Technical Assistance Agreement (TAA) was approved by the U.S. State Department on 6 October 2005. The amended TAA expands the technical scope to cover the research areas added under the ISP Phase II program, adds two dual citizens at UniMelb, and also covers Raytheon, Australia. The amended TAA has been signed by all the parties and is in force.

In support of the mathematical and information theoretic processing evaluations, Nick Coult (FMAH) spent a week at Raytheon in 10-14 April 2006 to evaluate the performance of FMAH's diffusion map signal processing approach on an set of Uncooled Infrared Imaging (UCIR) imagery collect under a Raytheon IRAD in support of an Air Force program. This data is ITAR restricted, company proprietary and considered sensitive by the Air Force Customer. Results of this preliminary evaluation can be provided separately as an appropriately marked addendum.

Other Accomplishments and Events:

- Received MATLAB code for two 1-Bit trackers from UniMelb on 21 April 2006.
- Raytheon personnel (Waagen, Stevens and Schmitt) visited Georgia Tech 3-5 April 2006 to discuss current CADSP hardware and algorithm status.
- Raytheon personnel (Waagen and Schmitt) and Georgia Tech personnel (Anderson) visited the group Dr. T.J. Klausutis (AFRL, Eglin) for a TIM on the

CADSP capabilities and to discuss possible AFRL laboratory interest in using the CADSP imager in their Optical Flow test program.

- Presented "Cooperative Control of Multiple UAVs for Passive Geolocation," at the Special Session on Cooperative Dynamic Systems, 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL.
- Attended the DARPA Waveforms for Active Sensing (WAS) Program Review Meeting 14-15 March 2006.
- Held a TIM on 15-17 March 2006 at ASU to discuss progress on their 1-bit detector for the distributed tracking demonstrations.
- Raytheon personnel (Waagen and Schmitt) visited UniMelb 15-23 April 2006 to discuss technical progress on ISP II.

1. E. Near Term Events

- Present "Comparison of Inter-class Divergence for Linear and Nonlinear Dimensionality Reduction, with and without Class Labels" and "Correlation of Inter-class Divergence and Classification Performance," at the *Combat Identification Systems Conference (19-22 June 2006, Orlando, Florida).*
- Obtain permission from the Air Force Program Office to release UCIR imagery to FMAH personnel. FMAH has the capability to process the UCIR imagery in an ITAR controlled environment and has a current Non-disclosure Agreement (NDA) with Raytheon Missile Systems.
- Release MATLAB simulation code used for the analysis in the "Cooperative Control of Multiple UAVs for Passive Geolocation," paper to Professor Daniel Pack, United States Air Force Academy. This code is understood to be at a research-level and used for that purpose.

2. A. Technical Progress

2.A.1. Raytheon Technical Progress

2.A.1.a. Distributed Sensor Demonstration

Wireless low-power sensor networks have gained much deserved attention in many research fields. With the advent of low-cost digital signal processors, wireless sensor networks have begun to emerge in many applications. The majority of military applications, including our particular choice of tracking of personnel though a field of distributed sensors, possess a common core of signal processing functions. Because such sensor networks will be laid down in an *ad hoc* configuration consisting of thousands of sensor nodes, accurate and scalable algorithms are critical. The algorithms and approaches that we are developing under ISP Phase II are thus expected to have wide applicability. For example, we are working closely with the Raytheon group that is demonstrating shooter localization under DARPA Information Exploitation Office (IXO) Networked Embedded System Technology (NEST) program. Self-localization is a significant computation challenge for NEST and an opportunity for technology transfer. .

For completeness, we again include a high level overview of the distributed tracking demonstrations. Detailed technical discussions and progress will be provided in several subsections that follow.

Distributed Tracking Demonstration Preliminaries

We divide the demonstration into a series of tasks that are critical for the demonstration and a set of tasks that would provide additional capability but are not critical to the accomplishing the demonstration. We refer to these latter tasks as "Extras." As illustrated in Figure 1, the distributed tracking demonstration consists of three functional blocks: (i) self-localization of the motes; (ii) 1-Bit on-mote detector; and (iii) base station tracker.

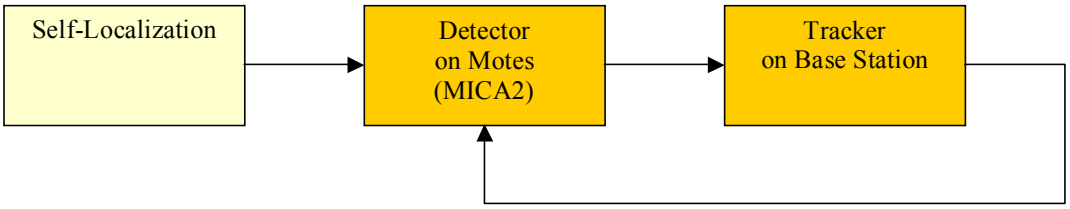

Figure 1: Demonstration Block Diagram

These three functional blocks are discussed in more detail below and flow into a schedule as shown in Table 1. The schedule shown in Table was presented at the last Quarterly Report, and it is updated to show the progress of each task. This schedule is divided into a series of tasks that are critical for the demonstration (depicted with an "M") and a set of tasks that would provide additional capability but are not critical to accomplishing the demonstration (depicted with a "B"). None of the non-critical tasks have been started as of today. However, almost all the critical tasks are either complete (depicted with a "C") or in progress (depicted with a "P").

Table 1: Distributed Tracking Schedule and Progress Summary

Task #	Task Description	System Component	Performer	From	To	Duration (week)	Bonus/Mandatory	Progress
1	Interface Specification Document (Matlab/nesC)	ISD	All	2/20/2006	3/3/2006	1.57	M	P
	Motes Self-Localization (Acoustic Ranging)	Localization	Thom/Sal	------	5/31/2006	2	B	
	Motes Self-Localization (RIPS)	Localization	Bob/Sal	------	5/31/2006	2	B	
	Motes Self-Localization Data Collection Improvement (RIPS)	Localization	Craig/UniMelb	------	5/31/2006	4	B	
	Motes Self-Localization GA Replacement (RIPS)	Localization	Bob/Michigan	------	5/31/2006	4	B	
	Motes Self-Localization Implementation/Evaluation (RSSI)	Localization	Bob/Michigan	------	5/31/2006	2	B	
2	Sensor Characterization (Microphone) Acoustic Model/ROC's	Detector	Sal/ASU	2/13/2006	2/27/2006	2	M	C
	Sensor Characterization (Accelerometer) Vibration Model/ROC's	Detector	Sal/Thom	------	4/17/2006	2	B	
	Sensor Characterization (Magnetometer) Magnetic Model/ROC's	Detector	Sal/Thom	------	4/17/2006	2	B	
3	Filter Acoustic Footstep (Microphone)	Detector	Sal/ASU	2/27/2006	3/20/2006	3	M	C
	Filter Vibration Footstep (Accelerometer)	Detector	Sal/Thom	------	4/17/2006	2	B	
	Filter Magnetic Noise (Magnetometer)	Detector	Sal/Thom	------	4/17/2006	2	B	
4	Energy Computation (Microphone)	Detector	Sal/ASU	3/20/2006	3/27/2006	1	M	C
	Energy Computation (Accelerometer)	Detector	Sal/Thom	------	4/17/2006	1	B	
	Energy Computation (Magnetometer)	Detector	Sal/Thom	------	4/17/2006	1	B	
5	Threshold (Microphone)	Detector	Sal/ASU	3/27/2006	4/3/2006	1	M	C
	Threshold (Accelerometer)	Detector	Sal/Thom	------	4/17/2006	1	B	
	Threshold (Magnetometer)	Detector	Sal/Thom	------	4/17/2006	1	B	
6	Transmit 1-bit Detection	Detector	Sal/ASU	4/3/2006	4/17/2006	2	M	C
7	Tracker Single Target (Particle Filter)	Tracker	Sal/Thom/ASU	4/17/2006	5/31/2006	6.28	M	P
	Tracker Unknown Number of Targets (Particle Filter)	Tracker	Sal/ASU	2/1/2006	5/31/2006	17	B	
8	Tracker (UniMelb)	Tracker	Craig/UniMelb	4/20/2006	5/31/2006	5.85	M	P
9	Detector/Tracker Integration	Integration	All	5/31/2006	6/30/2006	4.28	M	P
10	Motes Localization (Survey)	Localization	Bob/Thom/Sal	11/27/2006	12/1/2006	0.57	M	
11	Full Dress Rehersal Test	Testing	Bob/Thom/Sal	12/4/2006	12/15/2006	1.57	M	

M -- Mandatory to guarantee success of the demo
B -- Bonus/Extra accomplishment (integrated in the demo if completed in time)
C -- Task Complete
P -- Task in-Progress

Self-Localization:

- Survey – If available self-localization algorithms do not produce accurate enough results, we should just localize motes by survey them.

- [Extra] Acoustic Ranging – VU algorithm currently gives reasonable results for inter-mote distance of 9 ft. For inter-mote distance higher than 9 ft, parameters need to be tweaked to reduce error in measured ranges.
- RIPS – The code needs to be installed onto MICA2's. We may require permission to obtain UNCLASSIFIED code since it was developed under NEST program. Once installed, we need to make measurements behind M09 and evaluate results accuracy. Accuracy should be better than Acoustic Ranging Algorithm.
  - Drawbacks
    - [Extra] Current scheduling during data collection is too exhaustive and time consuming to make this a practical algorithm. For example, for only 16-mote network, data collection takes anywhere from 30 to 40 minutes. UniMelb wants to take this problem to improve scheduling by making only necessary measurements.
    - [Extra] Once measured data is collected, motes are localized using a Genetic Algorithm (GA). GA's are known to be computationally intensive (thus, slow to converge to a solution), and they do not always converge. UM will investigate replacing the GA with a more reliable and faster algorithm such as the steepest descent.
- [Extra] RSSI – Determine the accuracy of this Received Signal Strength (RSS) algorithm from UM. Also, make sure UM can implement it on MICA2's.

Detector (1-bit):

We next briefly discuss our detector development. The baseline demonstration will use a 1-Bit detector (target detected or not). This choice of detector implementation is driven by the network being so constrained in its communication capability. As shown in Figure 2, the detector is composed of four functional blocks: (i) Sensor Characterization; (ii) Signal Filtering; (iii) Energy Computation; and (iv) Threshold Calculation. For our scenario or tracking people through the network, sensor characterization consists of developing the acoustic signature of footsteps. Filtering is next performed to improve the Signal to Noise Ratio. A threshold is then set to produce the 1-Bit output of target detected or not. These four functional blocks are discussed in more detail below and again flow into a schedule as shown below.

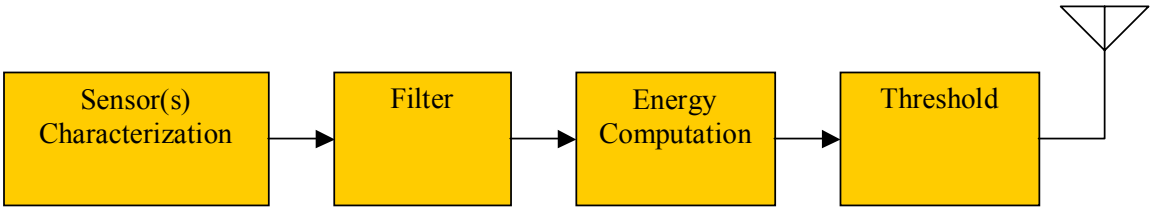

Figure 2: Detector Block Diagram

- Sensor(s) Characterization
  - Microphone (acoustic)
    - Person Walking – Determine ROC's to determine detector parameters, and motes network topology.
  - Accelerometer

        - [Extra] Person Walking – Determine if the sensor is capable of sensing vibration above noise floor on outdoor ground. If so, determine ROC's to determine detector parameters, and motes network topology.
    - Magnetometer
        - [Extra] Person Walking with Metal/Cell Phone – Determine if sensor can sense Metal or Cell Phones magnetic field. If so, determine ROC's to determine detector parameters, and motes network topology.
- Filter
    - Microphone (acoustic)
        - Person Walking – Develop Digital Filter similar to VU Acoustic Ranging Algorithm. The filter needs to be a Low-Pass. ASU will determine the frequency range of the filter.
    - Accelerometer
        - [Extra] Filtering can be ignored since vibrations are assumed to come only from people walking for the demo.
    - Magnetometer
        - [Extra] Filtering can be ignored unless interference from Earth Magnetic Field or Magnetic Noise in the area affect detector.
- Energy Computation
    - It can be extracted or be similar to VU Acoustic Ranging Code
- Threshold
    - Determine threshold based on ROC's.
- Transmit Detection.
    - 1 – Target Detected
    - 0 – Target Not Detected

Tracker:

- Received Data – Receive detected/not detected data from each mote.
- Track – Track target using:
    - Particle Filter
    - Australian Tracker

Software:

- Matlab – Use Matlab to integrate Demo components and display tracker's graphics/results.
- NESC – Use to implement Localization and Detector algorithms on MICA2's.

Interface Specification Document:

We have developed that first version of an Interface Specification Document describing the signals, variables, *etc.,* needed at the interface of each component of Figure 1. Input was received from ASU, UM and UniMelb. This document describes, for example, the data and signals that detector needs to provide to the tracker.

Summary:

Figure 3 represents the final demonstration. It will have 40 to 100 sensors (S_1, …, S_n) detecting a target and possibly multiple targets. There will be one or more Base Stations depending on the number of available trackers. The Base Stations (trackers) will graphically show the target location using Matlab interface

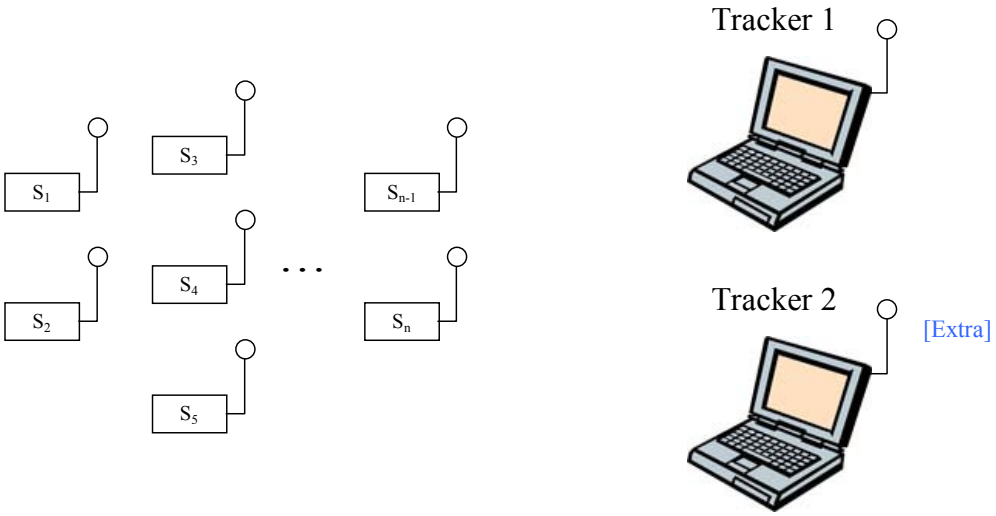

Figure 3: Demonstration Representation

Test Bed for the Distributed Tracking Demonstration

We have completed a test bed for the distributed tracking demonstration. This test bed gives the capability to easily integrate and test a variety of wireless sensor detectors and trackers. The main components of this test bed are shown on Figure 4, and they are the Detectors on the Wireless Sensors (Motes), the Base Station, and the Processing Station (Laptop).

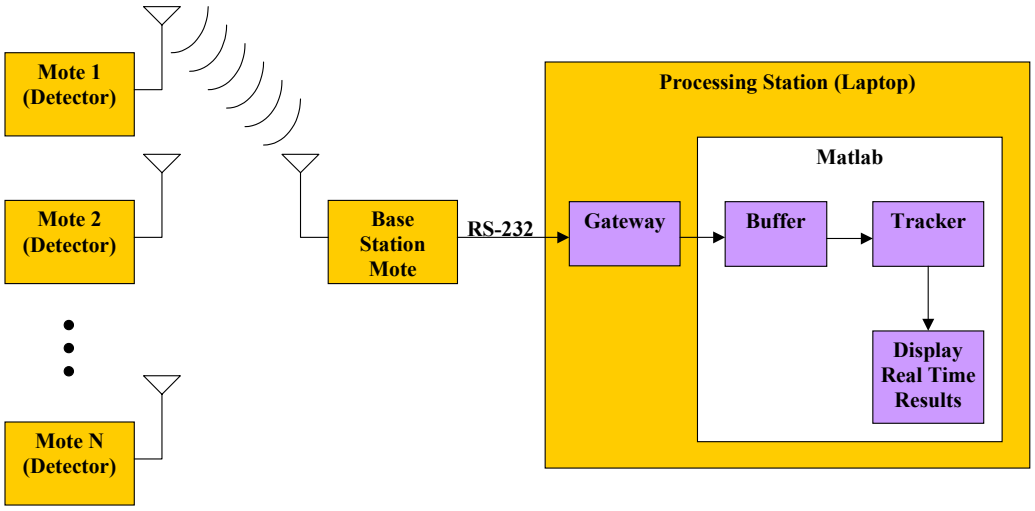

Figure 4: Test Bed Block Diagram for the Distributed Tracking Demonstration

The detectors on the motes can be any detector that can be implemented on the TinyOS and that can work with the limited hardware resources. The base station is a mote that receives the messages from the remote motes and transfers them to the serial buss RS-232 for input to the Processing Station (Laptop). The Processing Station consists of a

Gateway, and MATLAB components. The Gateway is a PC component that interprets the TinyOS message for MATLAB. The data, such as the Mote ID number that has detected a target, from the Gateway gets stored in a Buffer, which is then used by the Tracker algorithm. Finally, the target tracking results are displayed real time.

On 4 May 2006, the test bed was tested in the parking lot behind building M09 at Tucson. Thirteen motes (detectors) were placed on styrofoam cups in a rectangular grid with interspacing of nine meters in one direction and four meters in the other direction. These detectors were programmed to detect a beeping sound of approximately 4.4 KHz. A Lego robot representing a beeping target was programmed to cross the mote field. A photo of the test bed setup is shown on Figure 5.

Figure 5: Tracker Test Bed: Motes, Base Station Computer, and Lego Robot

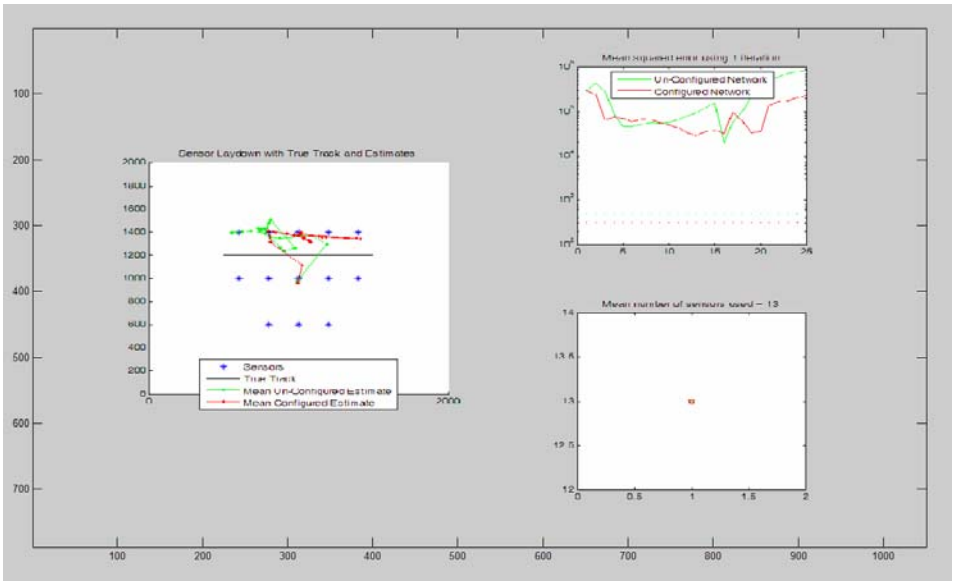

Figure 6: Base Station Computer displaying real time tracking results

While the motes are detecting the target, the real time results are being graphically shown on Figure 6 by MATLAB. Although the results shown on Figure 6 are inaccurate in that the target was not tracked accurately, the test bed itself worked well. Thus, to improve these results, the tracker and the detector parameters need to be adjusted.

2.A.1.b Coordinated UAV Geolocation of a Stationary Emitter

The geolocation problem is, simply stated, to discover the unknown location of a RF emitter. For purposes of this paper, simulated measurements used for estimation of the coordinates of the emitter consisted of time-differences-of-arrival (TDOA) of a signal at a number of different sensors. The sensors are assumed to be mounted on unmanned aerial vehicles (UAVs). The solution method described in our paper has two novel features.

It is well-known [Foy 1976] that positioning of the sensors has an effect on the solution, with accuracy of the solution being increased or diminished, respectively, by well or poorly placed sensors. Obviously, in the geolocation scenario, it is not known where to place the sensors because the location of the emitter is not known. All is not lost, however, as the first novel aspect of our approach is to move the UAVs after taking each measurement so as to be better placed when the next measurement is taken. The covariance of TDOA measurements is defined as $(\mathbf{A}^T\mathbf{A})^{-1}$, where

$$\mathbf{A} = \begin{bmatrix} \frac{x_1 - x_E}{R_1} - \frac{x_2 - x_E}{R_2} & \frac{y_1 - y_E}{R_1} - \frac{y_2 - y_E}{R_2} \\ \frac{x_1 - x_E}{R_1} - \frac{x_3 - x_E}{R_3} & \frac{y_1 - y_E}{R_1} - \frac{y_3 - y_E}{R_3} \end{bmatrix}$$

with (x_k, y_k) being the position of the k^{th} sensor, and R_k the (estimated) range from this sensor to the (estimated) position of the emitter at (x_E, y_E). (This is the matrix with three sensors; the matrix has $N-1$ rows if the number of sensors is N.) The magnitude of the covariance in each direction depends, in part, on the positions of the sensors. Thus, each UAV can be directed to move to the position that will minimize the trace of the covariance, constrained by how far the UAV can move during the allotted time interval.

There is thus a sort of "positive exchange" between sensors and processing, wherein the sensors take a measurement, process it, then move in the direction which the processor has indicated would be most optimal for taking the next measurement. Our simulations show that, as time progresses, the UAVs find themselves increasingly better positioned with respect to the emitter.

The second novel feature of our approach is use of the so-called "unscented" Kalman filter (UKF) for updating the emitter position estimate. The UKF has been around since about 1997 [Julier 1997] but, to our knowledge, has not previously been combined with TDOA for geolocation. Let us give a brief tutorial and show how the UKF is used in our application.

Kalman filtering works with two equations, a state evolution or "process" equation

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + v_k, \quad (1)$$

where $\mathbf{x}_k$ represents the estimate of the state at the k^{th} step, and a measurement equation,

(2) $$\mathbf{z}_k = g(\mathbf{x}_{k-1}) + w_k ,$$

which returns the k^{th} measurement, $\mathbf{z}_k$, that should be observed given the state. In equations (1) and (2), $\{v_k\}$ and $\{w_k\}$ are zero-mean, Gaussian noise sequences. If the functions f and g in these equations are linear (*i.e.,* matrix multiplication) then we use the by-now standard Kalman filter algorithm. If these functions are not linear we must resort to other means. The "other means" in this case is the following.

For the geolocation problem with a stationary emitter, the state is constant, thus the function f in equation (1) is merely the identity (we also ignore the noise term here). Following customary notation in Kalman filtering, let $\mathbf{x}_{k-1/k-1}$ denote the state estimate at step $k-1$ of the algorithm, and let $\mathbf{x}_{k/k-1}$ denote *prediction* of the state estimate at step k, based on the process equation (1). Then, in our application, we replace (1) with the following very simple process equation,

(3) $$\mathbf{x}_{k/k-1} = \mathbf{x}_{k-1/k-1} .$$

We next turn our attention to the measurement equation. Function g here is nonlinear, and what is done in the UKF is to generate a set of test points, the mean of these points being mean equal to $\mathbf{x}_{k-1/k-1}$, and their covariance equal to the current estimated covariance, which we denote by $\mathbf{P}_{k-1/k-1}$. The predicted measurement in (2) is then taken to be the mean,

(4) $$\mathbf{z}_{k/k-1} = \sum_{i=1}^{m} w_i \ \mathbf{z}_{k/k-1}^{(i)} = \sum_{i=1}^{m} w_i \ g\left(\mathbf{x}_{k/k-1}^{(i)}\right),$$

where $\mathbf{x}_{k/k-1}^{(i)}$ for $i = 1,2,\ldots,m$ denote the test points, and $\{w_i\}$ are weights. In our application, the measurement function is

$$g(\mathbf{x}) = \frac{1}{c}\begin{bmatrix} \|\mathbf{p}_2 - \mathbf{x}\| - \|\mathbf{p}_1 - \mathbf{x}\| \\ \|\mathbf{p}_3 - \mathbf{x}\| - \|\mathbf{p}_1 - \mathbf{x}\| \end{bmatrix}$$

(assuming three ships), where c denotes speed of propagation of the signal, and $\mathbf{p}_j$ denotes the position of the j th ship. Thus, function g predicts TDOA measurements.

The Kalman gain matrix is computed as follows,

$$\mathbf{K} = \mathbf{Q}\mathbf{S}^{-1}$$

where

$$\mathbf{Q} = \sum_{i=1}^{m} w_i \left[\mathbf{x}_{k/k-1}^{(i)} - \mathbf{x}_{k/k-1}\right]\left[\mathbf{z}_{k/k-1}^{(i)} - \mathbf{z}_{k/k-1}\right]^T$$

and

$$\mathbf{S} = \sum_{i=1}^{m} w_i \left[\mathbf{z}_{k/k-1}^{(i)} - \mathbf{z}_{k/k-1}\right]\left[\mathbf{z}_{k/k-1}^{(i)} - \mathbf{z}_{k/k-1}\right]^T + \mathbf{R},$$

where $\mathbf{R}$ is the measurement noise covariance. (An estimate of this covariance must be supplied to the filter.) The update, or correction, is now made, which is given by

$$\begin{aligned} \mathbf{x}_{k/k} &= \mathbf{x}_{k/k-1} + \mathbf{K}\left(\mathbf{z}_{k/k} - \mathbf{z}_{k/k-1}\right) \\ \mathbf{P}_{k/k} &= \mathbf{P}_{k/k-1} - \mathbf{K}\mathbf{S}\mathbf{K}^T \end{aligned} \tag{5}$$

In the first of equations (5), $\mathbf{z}_{k/k}$ denotes the actual TDOA measurement made at step k of the procedure. In the second of equations (5), $\mathbf{P}_{k/k-1}$ denotes the predicted covariance which, because of our assumption of a stationary state, is simply equal to the covariance at the $k-1$ step, that is

$$\mathbf{P}_{k/k-1} = \mathbf{P}_{k-1/k-1},$$

which is analogous to equation (3). We note that, in our application, the only place that a nonlinear equation is invoked is in equation (4).

In spite of its simplicity, the UKF performs rather well, at least in our simulations. We have compared it with a sequential batch least-squares method, and found that the UKF is far more robust to measurement noise than is the batch processing. See Figure 7. This seems to be due to the fact that the UKF, like any Kalman-type filter, has the ability to automatically "de-emphasis" measurements which are excessively degraded by noise, at least once it has progressed far enough to have sufficient history at its disposal, since it is then able to recognize a measurement which is too far from the mean. The least-squares method, on the other hand, has no automatic facility for doing this, and suffers.

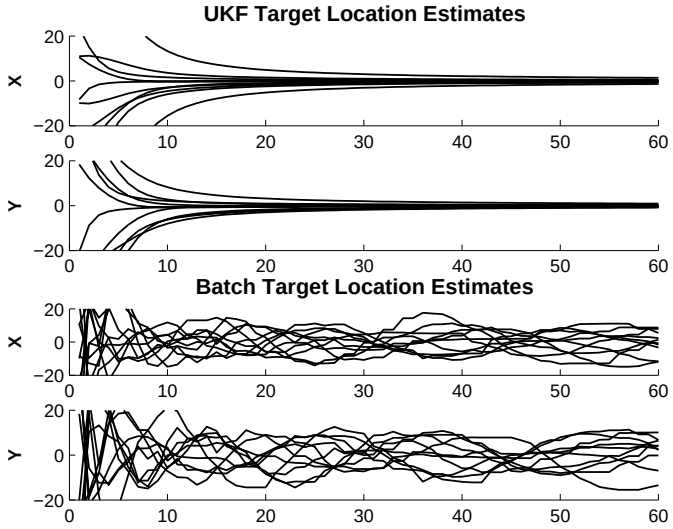

Figure 7: The figure shows convergence of the location estimate for sixty time-steps

In Figure 7, results of ten runs are superimposed. The top images are x- and y-coordinates of the estimate computed by the unscented Kalman filter algorithm, while the bottom two images are x- and y-coordinates of the estimate computed by the sequential batch processing, which uses a Levenberg-Marquardt least-squares solver. The true target position is at (0, 0). Simulated random noise of 50 ns standard deviation, normally distributed, was added to the TDOA measurements, which translates to an uncertainty of 15m on the ground. It can be seen that the batch processing estimate oscillates strongly within the noise bounds of ±15 m. In sharp contrast, the UKF estimate always coverges smoothly to the correct result, probably because the noise is Gaussian and Kalman filters

handle this case well. Nevertheless the UKF delivers superior results compared with a more straight-forward least-squares approach for this scenario.

In summary, this method for geolocation of a stationary emitter is working rather well in our simulation code. However, the method has only been tested with simulated data, and will require testing with measured data before we can make definite conclusions. We are working closely with another Raytheon team that is under contract with the Advanced Tactical Targeting Technology (AT3) program. This team has measured data available, as geolocation is an important part of the AT3 program, and eventually we will give them our code or they will give us their data for further testing. Eventually we hope that someone, either Raytheon or perhaps the Air Force Academy, will put this code on a vehicle and test it in the field. In the meantime, we are working to extend our methodology to include moving targets, a problem which is of interest to both the AT3 program and the Air Force Academy efforts.

2.A.1.c Evaluation of High Dimensional Data Processing Approaches

Modern high-performance, agile sensors are capable of producing tremendous amounts of data. Examples include polarization- and/or waveform-agile high resolution synthetic aperture radar (SAR) systems, multispectral systems, etc. Each temporal snapshot can be regarded as a high-dimensional vector. For statistical stability and processing throughput considerations in time-critical applications such as automatic target recognition, classification and identification (ATR), it is common practice to project the input high-dimensional data into an informative low-dimensional subspace, either by forming domain-expert-defined features or by applying data-driven dimensionality reduction algorithms.

In the case of data-driven dimensionality reduction, linear or nonlinear approaches can be selected to maximize either representation/parameterization efficacy or separability of class means. Examples include Principal Component Analysis (PCA) – linear/representation, Fisher Linear Discriminant Analysis (FLDA) – linear/separation, Laplacian Eigenmaps (LE) – nonlinear/locality preserving parameterization, and an extension to LE known as Class Constrained Dimensionality Reduction (CCDR) – nonlinear/locality preserving parameterization + separation.

In the ATR setting, classifier performance is commonly used as one of the metrics for optimizing sensor parameter settings, and as the main metric for optimizing feature selection. This process entails training and testing different classifiers, each having some set of tunable parameters, leading to a search space that is both computationally prohibitive as well as potentially obfuscatory of the inherent interclass separability. As an alternative to evaluating classifier performance, we apply a nonparametric divergence estimator (Henze-Penrose Divergence) for measuring relative interclass separability. We have performed two investigations in this regard. In both cases, the data is a subset of the well-known, multi-class MSTAR X-Band SAR dataset.

First, in a train/test setting, we evaluate the relative performance of PCA, FLDA, LE and CCDR applied to twelve targets. For the figure-of-merit, we use the mean of the pairwise interclass Henze-Penrose divergences of the test samples in the low-dimensional subspace (analogous to ranking classifier performance using ‘percent correct classification’). As features, we use the first ten subspace coordinates in each method.

Second, for a two-target case, we measure relative separability of subspace projections of the training data as well as the corresponding subspace projections of the testing data. We also apply three classifiers (support vector machine with optimized hyperparameter, *k*-nearest neighbor, linear) to the corresponding test subspace projections. We compare classifier performance with interclass train subspace divergence estimates, and we compare interclass train subspace divergence estimates with the corresponding interclass test subspace divergence estimates. This quantifies the predictive power of measuring interclass divergence in train data. As feature sets, we use all 1024 combinatorial subsets of the first ten subspace coordinates produced by PCA or LE.

The underlying data and derived results are ITAR-restricted. Therefore they will not be discussed here. The results will be made available to the DARPA Program Manager and ONR COTR at the next program review (June 13-14, Litchfield Park, AZ). The two investigations will also be reported as follows at the 2006 Combat Identification Systems Conference (June 19-22, Orlando, FL) in a U.S. – only setting.

2.A.1.d. CADSP UCIR Evaluation Technical Support

There is currently a great deal of interest in UCIR sensors for Automatic Target Acquisition (ATA) on smart munitions, such as the NetFires NLOS PAM. The Georgia Tech CADSP imager has the potential for being incorporated into *on-Focal Plane Array* (FPA) pre-processing operations; these include: Non-Uniformity Compensation (NUC) and non-linear/non-local pixel equalization. Traditional equalization approaches (*e.g.,* histogram equalization) tend to perform very poorly and it is likely that a localized, non-linear equalization approach is needed. Given ISP Phase II funding constraints, we will limit these pre-processing investigations to an evaluation of their implementation on the Georgia Tech CADSP imager. We have also been in preliminary discussions with Eglin, Air Force Base about using their optical flow test facilities and have already made a site visit to discuss this option.

2.A.2. ASU Technical Progress

2.A.2.a. Tracking Algorithms for the CADSP Configurable Imager

Our efforts during this reporting period have been primarily directed at extending our previous work to develop a multiple target tracker that will track targets in a scene as they enter, traverse, and then leave the surveillance region. The tracking is performed based on data received from the CADSP chip, which is configured according to the instructions of the tracker.

Our multi-target tracking algorithm is composed of two main parts: the CADSP chip controls and the target tracker. The CADSP controls determine the type of filtering operation that is performed on an 8x8 pixel block on a current frame of the tracking video sequence by the CADSP imager as it acquires the video. Moreover, it requests only the blocks in a frame that are relevant to the currently tracked targets and detection of new targets. The tracker incorporates the video blocks provided by the CADSP as measurements into the particle filter.

Currently we are working with real data acquired by a web-camera and processed to simulate the CADSP chip. Considering the controls part, we have developed an algorithm to select the blocks to be processed by the simulated CADSP chip that chooses

the minimum set of blocks necessary to assess the validity of each hypothesis made by the particle filter. This reduces the required information and computational expense dramatically compared to operating on the entire frame at each time step. Moreover, the selection of filtering operation (Gaussian or Mexican hat filter) is now indicated by a model designed to track human figures. The tracking algorithm is adaptive to the scene at hand by collecting training data that are used to construct foreground (targets present in the scene) and background (targets absent from the scene) distributions. These distributions are used as prior information throughout the tracking process. The tracking algorithm has been designed to deal with the occlusion of targets and implement constraints that do not allow the simultaneous tracking of a single target by more than one partition.

Figure 8 shows a sample frame of the video sequence used for tracking. It also indicates the model used for tracking human figures (white points). Each of these points represents an 8x8 pixel block. The block filtering operation to be performed is decided by the position of the point on the model. Presently, we are completing the multiple tracking code for tracking a known number of targets and exploring track-before-detect schemes for tracking newly arrived and leaving targets.

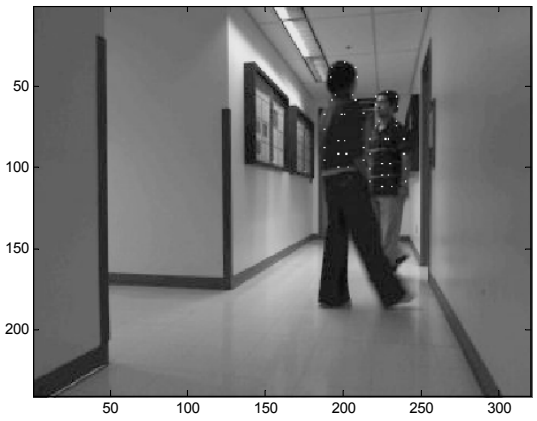

Figure 8: Video sequence used for tracking

2.A.2.b. Mote Tracking Support: Detector Development

Our efforts during this PoP were primarily directed at developing reliable and efficient methods of collecting acoustic data with the motes. After investigating several possible solutions, including several attempted modifications to the mote hardware, we have developed a viable solution by bypassing the motes' radio communication circuitry and transmitting data via wire using the motes' RS-232 interface. Using this solution, we have collected and analyzed a preliminary set of acoustic data, and analyzed the performance of the energy detector for it. The data was collected with a person walking on brick wearing tennis shoes, which made a fairly loud sound to provide one good benchmark case that could be used to compare to other surfaces.

The data was collected continuously for one-two minute intervals at various distances. The footsteps were recorded using one of the motes with the same ideal acquisition circuitry that will be used in the final implementation, and its data was transmitted directly to the base station computer over an RS-232 cable and saved for later

analysis. The sampling frequency was approximately 3.1 kHz, at 8 bits. This sampling frequency was chosen based on analysis of high-quality recordings done previously using a computer. Representative graphs showing data collected at eight feet are shown in Figures 9 and 10.

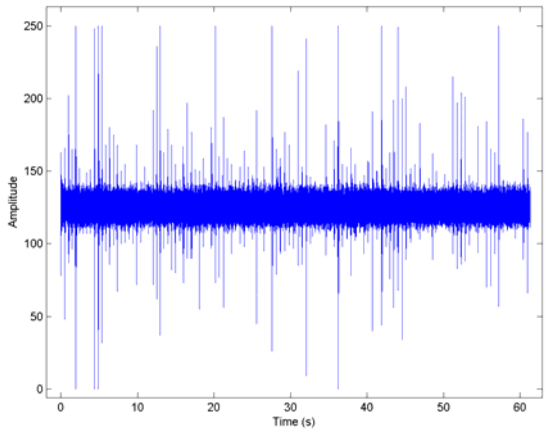

Figure 9: Time domain plot of acoustic data collected at 8 ft

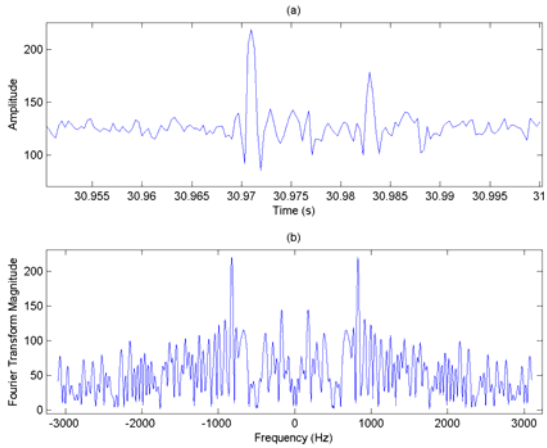

Figure 10: Time (a) and frequency (b) domain plots of one footstep at 8 ft.

After collecting the data, the probabilities of detection and false alarm were computed by simulating the energy detector with MATLAB. The energy detector computes the energy in a time window whose length was chosen to be about .1s; this was determined to be the typical duration of a footstep. The energy is given by $\sum_{k=t-n}^{t}|x_k-\mu|^2$, where x_k is the data at time k, μ is the mean of the data, n is the window size, and t is the starting time sample of the window. The detector bases its binary decision on whether this value is above or below some pre-set threshold. It then reports back to the fusion center, retrieves one new sample, and calculates the energy of the new window. This process is shown in Figure 11.

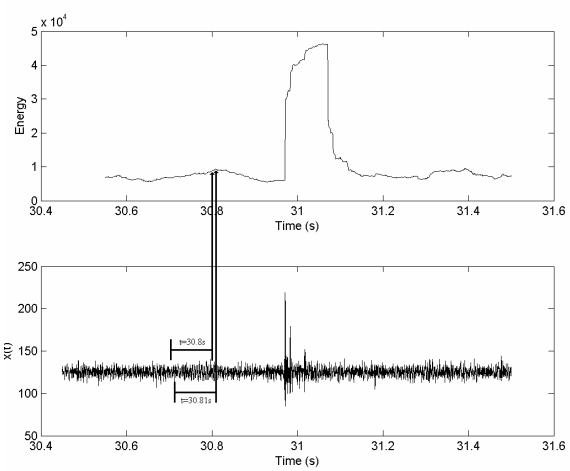

Figure 11: Energy calculations of two time windows at t=30.8s and t=30.81s

The threshold of the energy detector was then varied. Detector decisions that a footstep was present are counted as either a correct detection or a false alarm. In order to differentiate the two, a separate high-quality recording of the target was recorded at 1 ft from the target. This recording was synchronized to the data collected from the mote. Because the second recording had a very good SNR, an energy detector could be used to obtain a near perfect detection and false alarm rate. The points at which a footstep was detected were noted, and then compared to the points at which the mote's energy detector declared a detection. A slight time misalignment error was also allowed; if a detection was declared within a certain time error, it could still count as a correct detection. Once the number of samples between the stop and start points is known from the alignment, the mote data is upsampled by the ratio of voice recorder samples to mote samples within that time period. Figures 12 and 13 show the beginning and endings of the recordings (after being manually aligned as well as possible). Figure 14 shows a couple seconds of the data towards the middle of the recording where the alignment should be worst. The probability calculation process is shown in Figure 15 for a simulation with data from 2 ft and a threshold of 25,100. This resulted in a probability of detection of 0.707 and a probability of false alarm of 0.278.

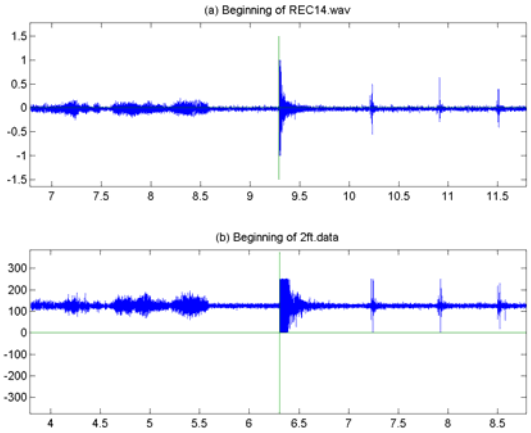

Figure 12: Alignment of the beginning of (a) the voice recorder, and (b) the mote data

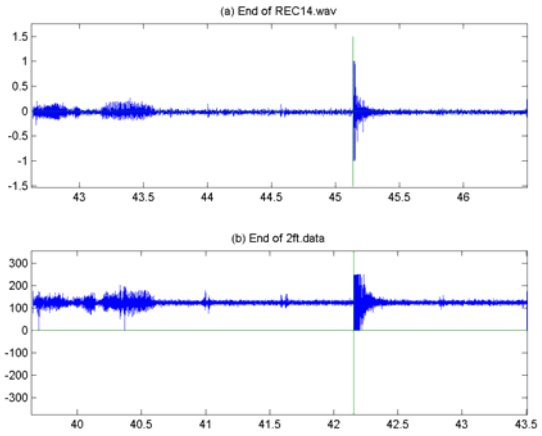

Figure 13: Alignment of the end of (a) the voice recorder, and (b) the mote data

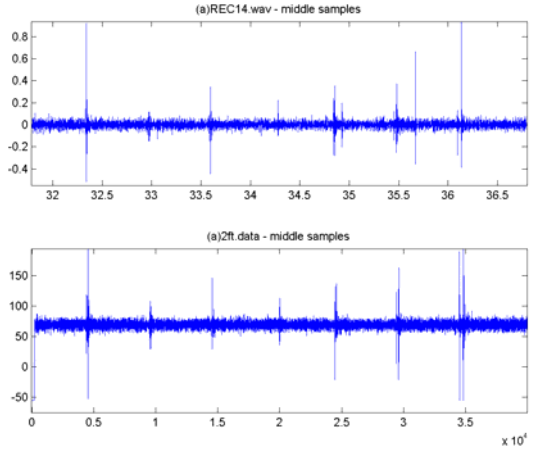

Figure 14: Alignment of the middle of (a) the voice recorder, and (b) the mote data

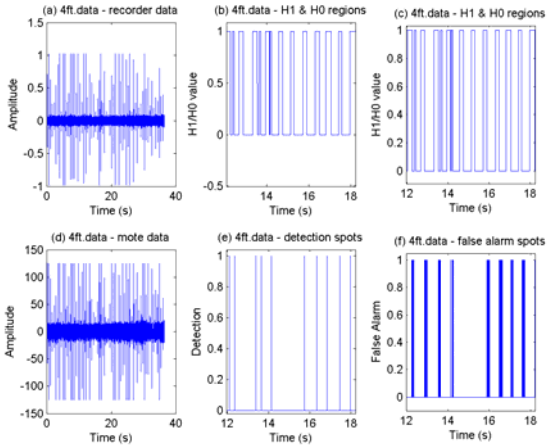

Figure 15: Time domain plots of: (a) recorder, (b) H1 region, (c) H0 region, (d) mote, (e) correct detection locations, and (f) false alarm locations for 4 ft

As can be seen in Figure 15, only one correct decision is counted per footstep. The probability of detection becomes the number of detections over the number of footsteps (the number of rectangular regions in the graph). For false alarms, any energy value over the threshold that was outside of a detection region was counted as a false alarm. Thus, the probability of error becomes the number of false alarms over the number of points outside the detection regions. The resulting ROC curve plots are shown in Figures 16 and 17. Note that the data at 6 ft is very out of place. After examining the data, there was a problem with the voice recorder, and the data did not match up well.

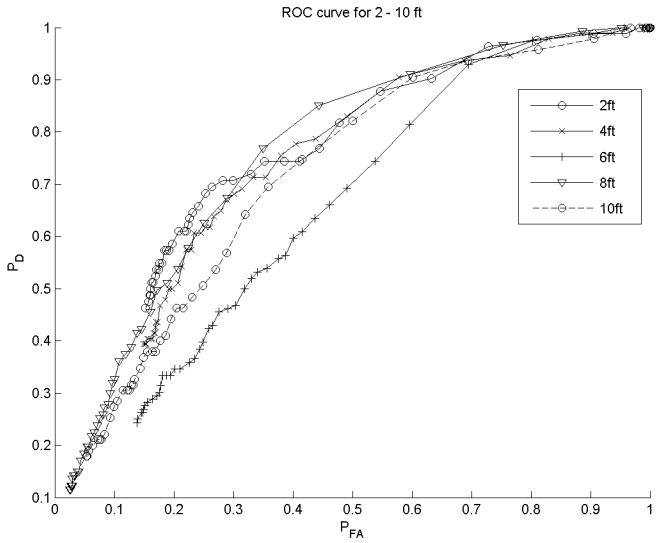

Figure 16: ROC curve for 2-10 ft

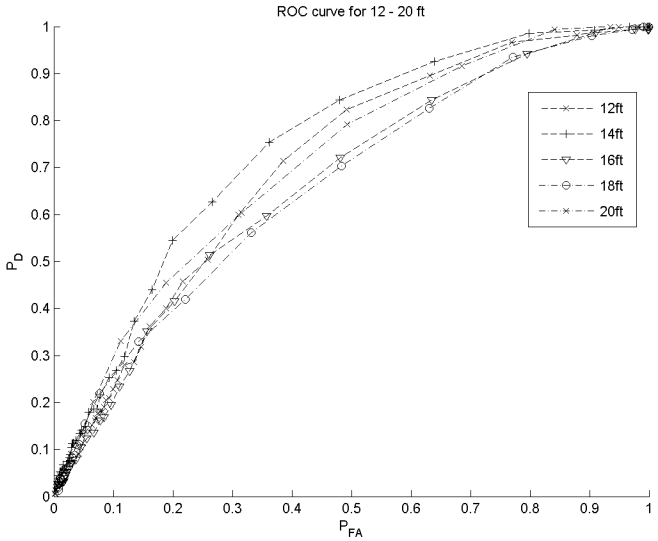

Figure 17: ROC curve for 12-20 ft

These preliminary results are promising. We are continuing to investigate filter structures that are matched to the footstep waveforms to determine if the detector performance can be improved through such structures. We have not progressed beyond

preliminary investigations. Also, the detection/false alarm points are being looked at to make sure that errors in alignment between the two recordings are not causing any misleading results.

Work is currently continuing to improve the results we have so far. We are looking into characterizing the mote detectors on different surfaces, such as asphalt and dirt, as well as different types of shoes, such as tennis shoes or boots. We are also trying to extend the work to a multiple-bit tracker, which uses the mote's energy values upon detection to compute the distance from the target to each mote. The tracker would then compute the target's position through triangulation. Work is being done now to verify an energy-*versus*-distance model that will be used in this tracker.

2.A.3. Georgia Tech Technical Progress

2.A.3.a Imager IC Development Status

A 256x256 version of the CADSP imager is currently under test. Temporary updates were made to previously designed boards to allow testing. As before, the testing interface includes MATLAB code with communication *via* ethernet to an FPGA motherboard which houses daughter boards of our custom design. Changes to FPGA code, C code, and MATLAB code were made to work with the newest imager. In the way of hardware, some modifications were made to adapt the old setup to the newest chip. Since the latest imager utilizes a 0.35 micron fabrication process instead of a 0.25 micron process, which has different voltage requirements, some changes were made to reference voltages on the previously designed boards. A new wire-wrap prototype board to interface the old board design to the newest imager was also constructed. This has sufficed for initial testing but once a comfort level has been reached with the design a new PCB will likely be designed.

Utilizing the modified setup, testing of the new IC has progressed. Initial testing if some analog circuitry utilized for on-chip bias generation was successful along with various mixed signal control structures. Initial control of the sensor array is also operational. Some difficulties initially arose in the control, digital noise problems, but they have been resolved by the addition of series resistors in clock lines and some modified digital sequencing. Here the intermediate wire-wrap board became very useful. With that under control, a non-transform low resolution *32x32* test image was read out in current mode via a intermediate access point. This was done by grouping pixels into blocks, of which there are 3*2x32*. Again the blocks are each 8 pixels x 8 pixels. Currently testing has moved to the sensing circuitry which follows the sensor array. The main operation here is a logarithmic I-V conversion. Using an on chip current source for testing, some initial transfer curves have been obtained that indicate proper operation. Right now effort is concentrating on getting good performance from these over as many orders of magnitude of current as possible. This directly affects the range of light the imager can process. The next step is to read an image using these I-V converters.

2.A.3.b Optical Flow Algorithm Status

The goal here has been to develop efficient algorithms for optical flow estimation that are closely tied to and accelerated by the CADSP imager. The algorithm on which we base this work is the LK-OFE (Lukas-Kanade Optical Flow Estimation) algorithm, which is based on weighted least-squares (WLS). To make fast algorithms, we can apply

recursive least-squares (RLS) techniques. This fast version can change $O(n^3)$ into $O(n^2)$ in the number of operations. As motion model is more complex, the number of operations saving increases. We are developing a fast gradient-based OFE using recursive least-squares (RLS) algorithm. Preliminary results indication that, by using RLS, the number of operation of WLS-based OFE algorithm can be reduced from $O(n^3)$ to $O(n^2)$. As motion model becomes more complex, the operational saving is expected to increase. The RLS-based fast OFE algorithm is anticipated to reduce the system requirements in the number of operations and power for designing optical flow estimation embedded system.

Future work in this area will focus on extending the 1-D window approach to a 2-D block-based RLS implementation. We will also be investigation ways to solve the double-sided exponential weighting case.

2.A.4. UM Technical Progress

In the three months since the last quarterly report we have made progress on three fronts.

2.A.4.a. Radio-interferometric Measurements for Self Localization

We introduced an indirect localization method for estimating sensor coordinates from Radio-interferometric measurements (RIMs) data. Localization from RIMs has suffered from very high computational complexity; however, this method provides a means for fast coordinate estimation. One of the main drawbacks of the new method had been its performance when RIMs have high-SNR. This quarter, we dramatically improved the high-SNR RIMs localization performance by incorporating an iterative feedback loop which adaptively improves the interim (pairwise distance) estimates [[Patwari & Hero 2006]]. This fast algorithm achieves 50 cm RMS location errors in high-SNR simulations. Further, its calculation can be partially distributed, and is a step towards a fully distributed RIM localization algorithm.

2.A.4.b. Progress on Received Signal Strength Localization Algorithm

We have been working with Crossbow Mica2 motes and TinyOS to implement accurate received signal strength (RSS) measurement and sensor localization platforms. We have extensive progress on both fronts. First, the RSS measurement test bed is complete and has been used in experimental measurement campaigns to record the time-varying multipoint-to-multipoint (M2M) radio channel. Additional automated programs have been developed to fully calibrate the sensor network to ensure accurate RSS measurements. We have used the network to provide experimental channel data to multiple projects, including RF motion detection, which has the promise of detecting human motion from behind walls. Finally, the measurements can be frequency-hopping and thus measure the frequency-variations of a M2M channel over time, or use the multiple channels for their frequency diversity which will improve RSS-based distance estimates. Finally, progress has been made towards fully distributed implementation of the dwMDS sensor localization algorithm, an effort which is now 75% complete.

2.A.4.c. CCDR Algorithm Progress

Classification constrained dimensionality reduction (CCDR) offers a way to combine dimensional reduction (*e.g.,* Laplacian eigenmaps) with label information. CCDR is often used as preprocessing tool for classification. To classify a new point x_{n+1} using CCDR, one needs to aggregate the point as an unlabeled point (*i.e.*, the class

indicator $c_{n+1}(k)=0$ for all k) to the collection of labeled training points $\{x_i\}_1^n$. Apply CCDR to obtain a lower dimensional embedding $\{y_i\}_1^{n+1}$. Then apply the classification of y_{n+1} based on $\{y_i\}_1^n$ and their respective labels. The resulting CCDR requires a GEVD of an $(n+K+1)\times(n+K+1)$ matrix (K is the number of classes) for every additional test point. We are interested in finding a low complexity out-of-sample extension (OSE) solution to this problem. Specifically, we would like to find a solution where we perform the $(n+K)\times(n+K)$ GEVD once and for a fixed small fraction of the cost of the GEVD perform the OSE for every new test point. Nystrom formula offers a solution by which the new test point is approximated by a weighted combination of the low dimensional embedding of the training points. We have implemented this form of the OSE and demonstrated (on the landsat data) performance, which is only slightly worse than that of the high complexity procedure described above. We are currently working on an OSE, which is based on small perturbation analysis of the GEVD.

2.A.5. UniMelb Technical Progress

2.A.5.a Tracker Algorithm #2 For Motes Sensor Network

In previous reports, we have described two different approaches we have taken to the problem of tracking targets using a field of motes. One of these is a hierarchical method based on the generation of virtual measurements followed by the use of clutter rejection tracking techniques to reject "false measurements" generated by the virtual measurement technique. The second is an algorithm that tracks directly from the data, and uses the unscented transformation to propagate distributions. During the last few weeks, Matlab programs that implement these algorithms have been sent to Raytheon to be incorporated into the mote demonstration. The algorithms have been coded so they can be used interchangeably as modules within the same program, that is, they have a common interface: inputs and outputs; thus facilitating comparison. Documentation has been provided within the code to explain the operation of the algorithms.

2.A.5.b Raytheon Technical Support

Research undertaken by Raytheon through its liaison with The University of Melbourne can be divided into four categories:

- Theoretical Scheduling
- Scheduling Passive Sensors for Geolocation
- Waveform Scheduling Against Smart Targets
- Random Projections for Radar

Details on these research areas follow.

Theoretical Scheduling

In previous reports, we have outlined analytic results for optimal scheduling of Gauss-Markov (GM) systems for terminal cost functions. Scalar systems were restricted to random walks, or

$$x_{k+1} = fx_k + w_k$$

with $f = 1$, and w_k a noise term, with only one of process noise covariance, initial state covariance, or measurement noise covariance varying across systems. Raytheon, in

conjunction with The University of Melbourne has worked on generalizing the results to include more general GM systems. During the quarter, we have developed preliminary results for f > 1; more details are forthcoming following a check of the results, and will likely be presented in a journal paper.

Furthermore, we have considered vector GM systems, with state estimates being computed with a fixed-gain filter. Corresponding analysis for using Kalman filters instead of fixed gain filters is complicated by the matrix inversion in calculating the Kalman gain.

We have considered a variety of *ad hoc* indexes for both scalar and vector systems in an attempt to analytically determine optimal scheduling; however, none of these have been optimal in a general setting.

Scheduling Passive Sensors for Geolocation

As previously mentioned, we are considering a novel form of geolocation using a laser rangefinder with passive secondary receivers. While we formulated the problem in the last quarter, we have begun to consider scheduling passive secondary receivers for tracking, instead of assuming that all sensors record measurements during each time step. Scheduling is performed by minimizing a cost

$$\chi = \alpha N + E[P_k]$$

where α is a cost for using N receivers, and $E[P_k]$ is the expected track error covariance. In our formulation, P_k is determined by using an unscented Kalman filter (UKF), as the covariance depends only on the measurement statistics, projected covariance, and sigma point selection scheme. While a myopic approach may not be strictly optimal, extending the horizon to greater than one suffers, as future P values do depend on measurements, through the dependence of a sigma point selection scheme surrounding the future state estimates.

Waveform Scheduling Against Smart Targets

One problem we address during ISP is the effectiveness of a waveform against a maneuvering target. To this end, we are trying to evaluate waveform performance against smart targets, in the assumption that the target can observe waveforms being utilized against it, and react accordingly. This yields a game in which the target, upon realizing that it is being observed, may adjust its behavior. The RF may detect this reaction, and modify its behavior. The cycle may continue *ad nauseum*.

We are currently investigating an idealized version of the problem, in which the target and sensor have a finite number of known strategies, of which they select a pure strategy for every engagement in a rational manner. Currently, we are investigating target models for a constant acceleration or constant turn, with a hypothetical sensor that is able to take position and velocity measurements. A mathematical game is formulated using an upper bound of estimated state error covariance using results from Sinopoli *et al.*

We realize that this is a rather idealized case, and hope to extend results in the future of ISP, and related work on WAS.

Random Projections for RF

In the presence of clutter, it has been proposed to use a minimum value measure of effectiveness (MVMOE). As mentioned, that measure can be difficult in practice due to the inversion of a generally ill-conditioned matrix. While that problem has been circumvented by projection of the matrix onto its relevant range space, we aim to solve the problem via random projection, which has a documented effect of effectively whitening the information. Whether or not this whitening removes the information that differentiates one waveform from another remains to be seen.

Furthermore, we aim to use the Danzig Selector (DS) as a possible means to identify the presence of targets. Unfortunately, the DS requires an unknown vector to be sparse in some known basis. It remains to be seen whether or not this assumption is applicable to general radar problems.

2.A.5.c Tracking with Motes

1. Scheduling for Distributed Passive Sensors

Recent advances in low-power micro-sensors and wireless network technology have led to an increase in the use of networks of distributed passive sensors. Such networks are typically made up of relatively cheap sensors, with limited processing resources and battery life. The limited on-board resources prevent significant computations being performed at the sensor level. Instead, they transmit information over bandwidth limited communications channels. However, typically communications consumes more power than either computation or sensing.

Common assumptions when tracking a target moving through a distributed sensor network include the assumptions that:

- all sensors are active at all times; and
- all information can be transmitted to some central processor.

In practice, both of these are unrealistic. An additional, common assumption is that the sensor network is homogeneous, i.e. that all the sensors are of the same type. In this part of the project we will design and evaluate the performance of a target tracking algorithm for a distributed network of passive sensors that relaxes all three of these assumptions.

The particular scenario we will consider is that of a single target moving through a field of heterogeneous sensors. The network will be made up of primarily acoustic sensors with an additional, smaller number of infrared (IR) sensors. The acoustic sensors are proximity sensors, i.e. they transmit a packet when they detect a target within their sensing range. The IR sensors are more sophisticated. In addition to detecting a target, they are capable of determining in which quadrant of their sensing range the detection occurred.

The tracking algorithm will operate adaptively. It will schedule which sensors, in terms of both location and type, are to be activated at the next scan, with the goal of minimizing an appropriate cost function. This cost function will compute the trade-off between resource usage (both battery power and communication bandwidth) and detection and tracking accuracy.

The tracker evaluation will be performed off line. Data will be collected from experiments of a target moving through the sensor field when all sensors are active and

all data is transmitted. The scheduling tracking algorithm will then select the appropriate subset of the collected data to form its tracks. The performance of the scheduled tracker can then be compared to the performance of a tracker which has the maximum amount of data available to it.

This work will consist of two major parts:

- the design of a tracking algorithm that uses a mix of acoustic, proximity sensors and IR sensors; and
- the design of a scheduled version of this tracking algorithm.

2. Computationally Efficient Mote Localization

2.1 Overview

In [Maroti *et al.* 2005] an algorithm is presented for performing relative localization of a network of N MICA2 motes using the radio interferometric positioning system (RIPS) technique. It requires O(N2) measurements to be taken and uses a genetic algorithm based method for computing the coordinates of the motes. The method assumes all motes can communicate with each other. The goal of this project is to develop a robust, computationally efficient method of localizing a large network of motes using the RIPS method. In a large network, the assumption that all motes are in range of each other will not hold.

Each RIPS measurement requires four nodes, *A, B, C and D*. Two nodes transmit at different, but relatively close frequencies, while the other two act as receivers. The resulting relative phase difference between the received signals is a function of the pairwise distances between the nodes. Figure 18 shows the case when *A* and *B* are the transmitters and *C* and *D* are the receivers. The resulting relative phase difference yields:

$$D_{ABCD} = d(A,D)-d(B,D)+d(B,C)-d(A,C) \quad (6)$$

where we call D_{ABCD} the RIPS measurement.

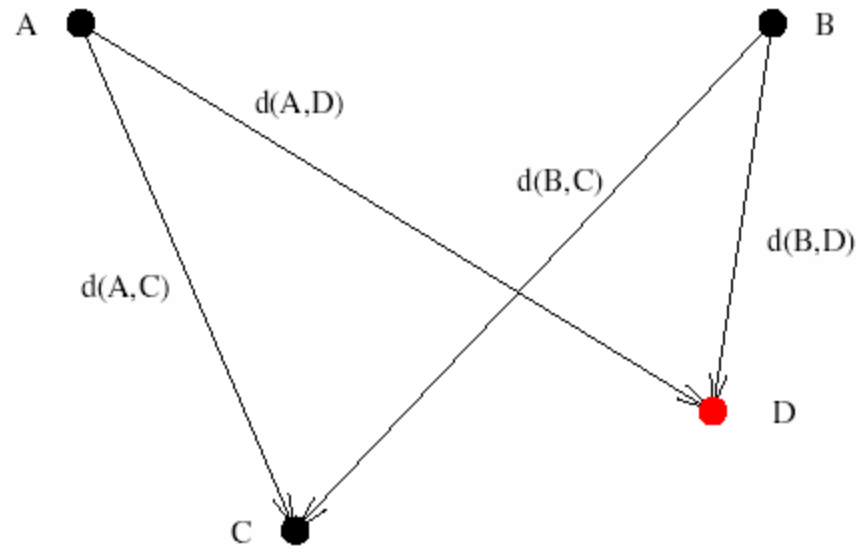

Figure 18: RIPS measurement process: A & B are transmitters; C & D are receivers

Similarly, when *A* and *C* are transmitters and *B* and *D* are the receivers the RIPS measurement is

$$D_{ACBD} = d(A,D)-d(C,D)+d(B,C)-d(A,B) \quad (7)$$

For a given set of four nodes, it can be shown that these are the only two independent measurements that can be made. That is, all other combinations of transmitter and receivers pairs yield measurements which are linear combinations of Equations 6 and 7.

2.2 Efficient Localization

Given knowledge of the location of three *anchor* nodes, *A, B* and *C* and two independent RIPS measurements we have developed a two-stage, closed-form method for determining the location of the fourth node D. This method is based on a similar approach in [Mellon *et al*. 2003] for localization using time difference of arrival (TDOA) measurements.

The first step requires the solution of a quadratic that calculates the distance between *D* and one of the anchor nodes. This result is then substituted into a set of two linear equations to determine the location of *D*. Thus, this method requires only *O(N)* RIPS measurements.

Given appropriate geometry, the quadratic equation in the first step will have a single, positive solution. This yields a unique location for the unknown node. In some cases, there will be two possible solutions. To resolve this ambiguity an additional RIPS measurement, using an additional anchor node, is required to localize *D*.

2.3 Effect of Noise in the RIPS Measurements

The localization technique outlined in the previous section will give the exact location of the unknown node when there is no noise. In practice, there will be uncertainty in the RIPS measurements and possibly also in the locations of the anchor nodes. In this section, we outline the preliminary studies we have done to investigate the effects of these sources of uncertainty. These results are all based on simulations. An analytical study of noise effects is part of on-going work.

In the simulations, we assume that the locations of the anchor nodes are known precisely but there is zero mean Gaussian noise in the RIPS measurements, with variance given by σ^2. Figure 19 shows the effect of this noise on localization performance when $\sigma = 0.1$. The actual location of each node that is to be localized is given by the blue dot. The blue ellipses indicate the uncertainty in the distance between each unknown node and node A as a result of noise in the RIPS measurements when *(A, B)* and *(B, C)* are the transmitter pairs. The green ellipses indicate the equivalent uncertainty when the transmitter pairs are *(A, C)* and *(B, C)*. The clear area around node A represents the area when there are two possible solutions for the distance between A and the unknown node, so it cannot be localized unambiguously.

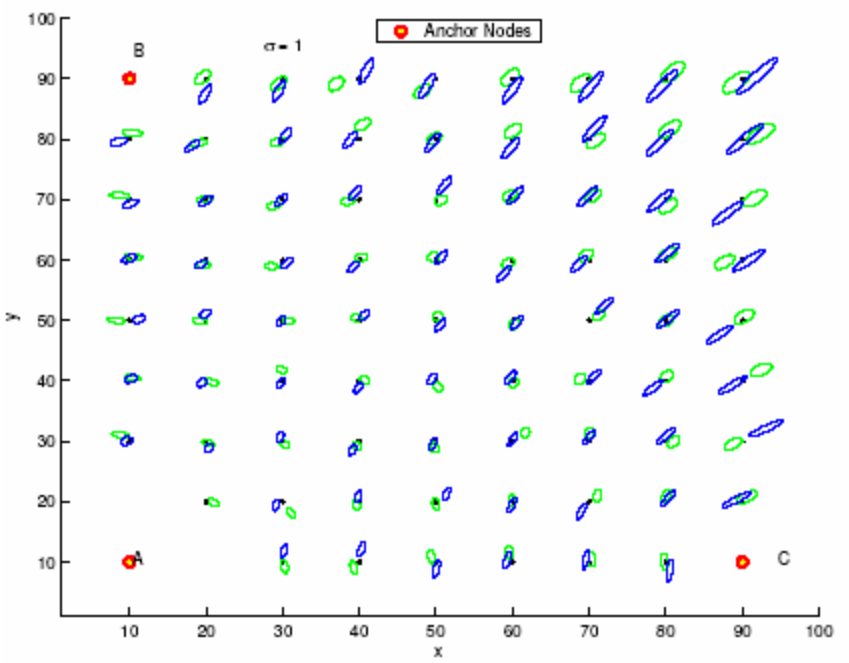

Figure 19: Uncertainty ellipses in computed distance between anchor node A and unknown nodes when there is noise in the RIPS measurements. Blue dots represent true node locations. Blue and green ellipses show the uncertainty in the true distance for two possible sets of transmitter pairs

2.4 Localization Propagation

The analysis above assumed all nodes are in range of each other. In practice, this will not be the case. Instead, once all nodes in range of the original three anchors are localized, some subset of these will then become anchor nodes in turn to localize more remote nodes. As a result, any errors in their estimated positions will propagate. Currently, we are investigating robust methods for localizing a large field of sensors, where the surveillance region is larger than the radar range, which will ameliorate the effect of noise propagation. This will be achieved by an appropriate choice of new sets of anchor nodes which have the best geometry and possibly by the taking of additional measurements.

2.5 Future Work

The method discussed in the previous section requires three anchor nodes in known locations with overlapping coverage areas. There are scenarios where such a distribution of nodes is reasonable, for example perimeter security of a relatively small area. However, for larger areas this is not the case. To deal with such scenarios we are investigating computationally efficient and robust solutions where anchor nodes are randomly located throughout the surveillance region.

Consider the case where a subset of nodes is equipped with GPS so that their locations can be measured accurately. Suppose these are distributed randomly over the surveillance region, but at a sufficient density that there is overlap between the coverage regions of adjacent anchor nodes and multiple nodes within that overlapping area. An example of this is shown in Figure 20. Here *A* and *D* are anchor nodes; *B, C, E* and *F* are pseudo-anchors; and *X, Y* and *Z* are nodes whose location is unknown. The coverage region of each of the anchors is shown by the dashed lines.

Suppose the location of each of the nodes in the overlapping area, *X, Y* and *Z*, are computed using the triple (*A, B, C*) as anchor nodes and then the triple *(D, E, F)*. Thus, for each node in the overlap there will be two possible locations which are functions of the unknown locations of the pseudo-anchors. We conjecture that it will be possible to localize the pseudo-anchors by combining these pairs for all three unknowns. Once the locations of the pseudo-anchors are determined, the location of all nodes in both regions can, in turn, be calculated.

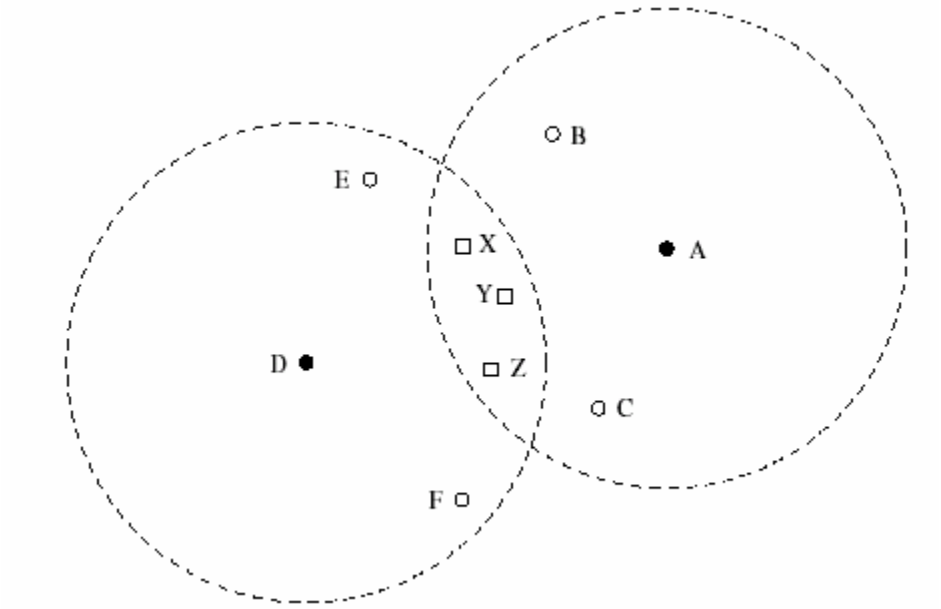

Figure 20: Distributed anchor nodes example. True nodes are given by closed circles, pseudo-anchors by open circles and nodes of unknown locations by squares.

2.A.5.d Waveform Scheduling for Maneuvering Targets

1 Introduction

We describe some extensions of earlier work on waveform scheduling for maneuvering targets. Our original work [Howard *et al.* 2004] on waveform scheduling has been extended by our colleagues at ASU (Papandreou, Morrell, Sira, [4]) to accommodate more suitable waveform libraries using nonlinear FM waveforms. Here we take the ASU waveform library and fit it into our tracking and cost function model. Simulations are done to show the effectiveness of waveform scheduling in this context, where effectiveness is measured in terms of determination of the dynamical tracking model and the RMS error of the tracker.

For the purposes of this report, we assume a single maneuvering target. The tracking is done using an Interacting Multiple Model (IMM) filter. Our cost function for a waveform is (an approximation to) the mutual information between the measurement and the dynamical model of the IMM.

2 Agile Waveform Tracking with IMM filter

We assume that the dynamical models and the sensor measurement processes are linear and described by the following equations

$$x_k = F(\Theta_k)x_{k-1} + \nu(\Theta_k)$$

$$z_k = Hx_k + \omega(\phi_k) \tag{8}$$

where the dynamical model Θ_k is a discrete random variable, which at time *k* can take any value $1,\ldots M$; $\phi_k = 1,\ldots N$ is the waveform used to obtain the measurements at time *k*. A number of generalized frequency modulated chirps form the library of waveforms. We write x_k for the state of the track and z_k for the measurement at time *k*. $F(1),\ldots F(M)$ are the state propagation matrices for the different maneuvers, H is the measurement matrix. Process noise is denoted by $\nu(1)\ldots\nu(M)$ and measurement noise by $\omega(1)\ldots\omega(N)$. These are all zero mean, white, and uncorrelated Gaussian noise sequences with covariance matrices $Q(1)\ldots Q(M)$ and $R(1)\ldots R(N)$ respectively. We assume that changes in target trajectory can be modeled as a Markov Chain with given transitional probability matrix P, *i.e.,*

$$P_{i,j} = P\{\Theta_k = j \mid \Theta_{k-1} = i\}; \quad i, j \in [1, M] \tag{9}$$

The trajectory of the target can be described at any time by one of the M dynamical models. The tracker switches modes between the dynamical models using the measurements, and thus facilitates tracking of maneuvering targets. Our problem is to choose the waveform which will minimize the entropy rate of dynamical model.

Our aim is to achieve better tracking performance with the IMM filter by scheduling the waveforms.

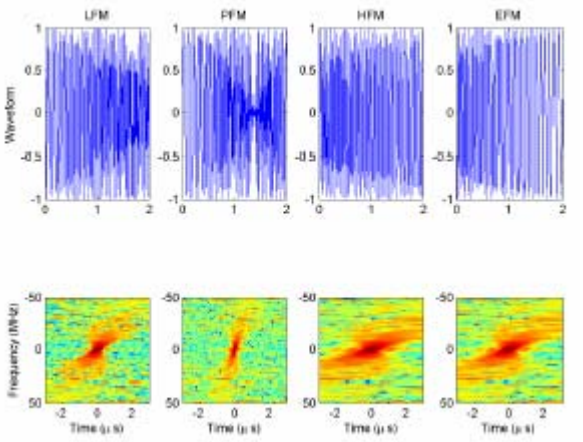

Figure 21: Upsweep Generalized Chirp Waveforms and Their Ambiguity Functions

3. Non-Linear FM Waveform Library

In these experiments, the linear chirp waveform library [Howard *et al.* 2004] is enhanced with generalized FM chirps, similar to the library in [Sera *et al.* 2006]. Along with linear FM up-sweep and down-sweep chirps we have considered power, hyperbolic and exponential FM up-sweep and down-sweep chirps. Each waveform can be represented as

$$s(t) = w(t)\exp(2\pi i \lambda f(t)) \tag{10}$$

where *w(t)* is a complex envelope (identical for all waveforms in the library), λ is a sweep constant and *f(t)* is a phase function:

$$f(t)=\begin{cases} t\Big/\gamma+\big(t+t_0\big)^2\Big/2 & \text{linear FM} \\ t\Big/\gamma+\big(t+t_0\big)^k\Big/k & \text{Power FM, k - real number} \\ \dfrac{-2t_0+\gamma}{2\gamma t_0}\log\big(t+\gamma+t_0\big) & \text{Hyperbolic FM} \\ \dfrac{-\gamma}{\exp\left(-2t_0\Big/\gamma\right)-1}\exp\left(-\big(\mathrm{t}+\mathrm{t}_0\big)\Big/\gamma\right) & \text{Exponential FM} \end{cases} \tag{11}$$

Figure 21 shows an example of such waveforms for positive λ (*i.e.*, up-sweep chirp) and their ambiguity function. Of course, to be complete, the library must consist of both up and down-sweep waveforms of these types. In high SNR the waveform covariance matrix can be approximated the CRLB on the estimation of delay and Doppler, which is obtained by inverting the Hessian of the Ambiguity Function, evaluated at the true target delay and Doppler [Van Trees, Kershaw & Evans 1997].

4 Cost Approximation

In this section we derive an approximate formula for calculating the cost $I(\Theta,Z)$ in the form of the expected Kullback-Leibler divergence from $P\{Z_k \mid \Theta_{k+1}\}$ to $P\{Z_k\}$. We have:

$$\begin{aligned} P\{Z_k\} &= \sum_i \mu_{k|k-1}(i)\Lambda_i^{\phi}(z_k) \\ P\{Z_k \mid \Theta_{k+1}\} &= \frac{\sum_i P_{ij}\Lambda_i^{\phi}(z_k)\mu_{k|k-1}(i)}{\sum_i P_{ij}\mu_{k|k-1}(i)} \\ &\sum_i B_{ij}\Lambda_i^{\phi}(z_k) \end{aligned} \tag{12}$$

where $B_{ij}=P\{\Theta_k=i \mid \Theta_{k+1}=j\}$ is the backwards transition probability.

In order to calculate the Kullback-Leibler divergence, we approximate $P\{Z_k \mid \Theta_{k+1}=j\}$ and $P\{Z_k\}$ by single Gaussians $\aleph(y_j,R_j)$ and $\aleph(y,R)$ respectively, where:

$$\begin{aligned} y_j &= \sum_i B_{ij}\hat{z}_k(i), \quad j=1,\ldots M \\ R_j &= \sum_i B_{ij}\left(S_k(i)+\left(y_j-\hat{z}_k(i)\right)^T\left(y_j-\hat{z}_k(i)\right)\right), \quad j=1,\ldots M \\ y &= \sum_i \mu_{k|k-1}(i)\hat{z}_k(i) \\ S &= \sum_i \mu_{k|k-1}(i)\left(S_k(i)+\left(y-\hat{z}_k(i)\right)^T\left(y-\hat{z}_k(i)\right)\right) \end{aligned} \tag{13}$$

and $\hat{z}_k(i)$ is the predicted measurement for ith model.

Using the formula for Kullback-Leibler divergence from a Gaussian to a Gaussian distribution, we obtain:

$$I(\Theta;Z) = E_{\Theta_{k+1}}\{D_{KL}(P\{Z_k \mid \Theta_{k+1}\} \| P\{Z_k\})\}$$
$$\approx \frac{1}{2}\sum_{i=1}^{M}\mu_i\left(\log\frac{|S|}{|R_i|} + tr\left(S^{-1}R_i\right) + (y-y_i)S^{-1}(y-y_i)^T - N\right) \quad (14)$$

for N-dimensional Z.

5 Experimental Results

We have considered tracking of a single maneuvering target in range-Doppler coordinates with five acceleration modes: -20,-10, 0, 10, 20 m/s^2 using IMM filter as in [Howard *et al.* 2004]. In the first set of experiments the waveform library consisted of linear maximum bandwidth up-sweep and down-sweep chirps and six max bandwidth up and down-sweep power FM chirps for k = 2.5, 3, 3.5. The up-sweep waveforms, their ambiguity function and measurement error covariance are shown in Figure 22. For down-sweep waveform ambiguity and error covariance are just the mirror image of the up-sweep chirps. The outcome of the experiment is presented in Figure 23. The second set of experiments was performed for libraries consisting of all described types of generalized FM. The result is represented on Figure 24.

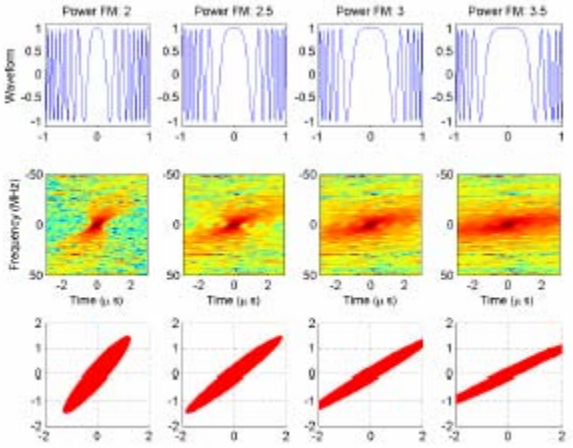

Figure 22: Upsweep Chirp Waveforms, Ambiguity Functions & Error Covariance

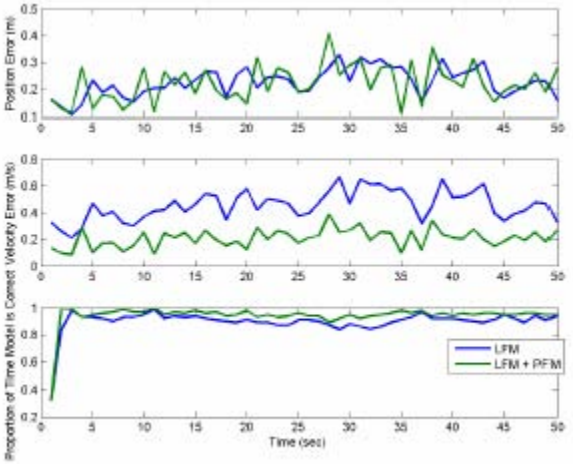

Figure 23: Tracker Performance Comparison: LFM and LFM+PFM waveform libraries

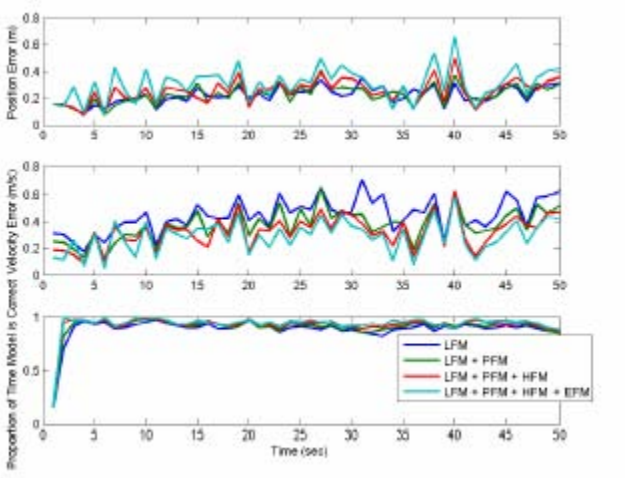

Figure 24: Comparison of Tracking Performance for LFM, LFM+PFM, LFM+PFM+HFM and LFM+PFM+HFM+EFM libraries of waveforms

6 Conclusions

The rather obvious conclusion is that the performance of the tracker improves with the additional waveforms. However, that improvement is not significant except perhaps in the velocity estimate emanating from the tracker. The PFM waveforms show significant improvement in this estimate; the addition of HFM and EFM waveforms appear, at least in these simulations, to overcome some losses in position estimation from the addition of the PFM waveforms alone. Clearly more simulation work is needed here over many different scenarios and with a variety of tracker models.

2.A.5.e Progress in Other Sensor Scheduling

Under ISP Phase II, we have been investigating two sensor scheduling evaluations. One research area is sensor scheduling for counter swarms, particularly for the TBM scenario. The second research area is sensor scheduling for passive emitter geolocation. We now discuss in more detail our approach and progress in these areas.

1 Sensor Scheduling for Counter Swarm

1.1 Introduction

Evaluation one explores gains that can be achieved by smart sensor scheduling when tracking a large number (swarms and TBM) of targets. By smart sensor scheduling we mean individual track update choice.

The sensor is assumed to operate in two modes. Mode one, or Surveillance Mode, is assumed to take a fixed percentage of time, in our example 20%. In this mode the sensor makes one sweep of the whole surveillance area, with the aim of finding new targets, as well updating all existing tracks. In mode two, or Track Update Mode, only a single potential target (track) is updated at a time, until the time for this mode runs out.

In both modes of operations, a low, but non-zero, clutter measurement density exists. In both modes of operations, targets are detected with probability of detection less than one. In this environment, automatic target initialization creates both true tracks and false tracks. True tracks are tracks which follow a target, usually, but not always, initialized using detections from this target. False tracks do not follow a target, one or more clutter measurement is usually involved in their initialization; or they may be

initialized by detections from different close targets in different scans. It is also possible for false tracks to become true tracks and vice verse. True track may become false either because the tracker has lost the target, or the target has physically ceased to exist.

Sensor scheduling is usually designed to minimize track estimation errors. Here we take another approach. We deem that the most important task is to determine the number of targets in the surveillance area and their approximate position, estimation error minimization is the second priority.

Each track can have two states. Newly initialized track is a tentative track. Each track recursively updates a track quality measure. If the track quality measure falls below a termination threshold, the track is deemed to be false and removed (terminated). If the track quality measure rises above a confirmation threshold, the track is deemed to be true and confirmed. This procedure is termed "false track discrimination".

The sensor thus reports the confirmed tracks as true tracks. As the tracker operates in a stochastic environment, the false track discrimination will result in errors, some true tracks will get terminated, and some false tracks will get confirmed. Sensor scheduling in Evaluation One aims to improve the false track discrimination properties.

1.2 Approach

A number of simplifying assumptions are employed, which: (i) will not change the nature of the results, e.g. proof of concept will be valid; and (ii) significantly simplify the software simulation. Once the proof of concept is obtained, the simplifying assumptions will be removed sequentially to obtain more realistic results. The simplifying assumptions are:

- Targets will move with uniform motion, corrupted by (some) plant noise.
- Targets are detected with constant and known probability of detection.
- Clutter measurement density is small but non-negligible. It is Poisson distributed with uniform spatial density.
- Sensor has infinite resolution.
- Sensor provides 2-D information: range and bearing.
- Target measurements can be linearized using the unbiased measurement conversion. Thus sensor measurement errors are assumed Gaussian in Cartesian coordinates, with covariance matrix a function of range and bearing.
- Sensor provides measurement position only; measurement amplitude and Doppler are not provided.
- Sensor works in two modes:
    - Surveillance mode, in which a sweep of the whole surveillance area is performed, including the areas covered in the Track Update Mode. This mode will occur at regular intervals, and will take a fixed percentage of total time.
    - Track Update mode, in which sensor visit designated tracks, one at a time, unless they are in close proximity to each other. It is assumed that each

update takes a fixed time interval, independent of the track position. It is also assumed that the probability of detection and clutter measurement density will be the same in both modes of sensor operation.

The system under consideration then consists of a sensor, tracker and Sensor Scheduling Control, where the sensor is as described above.

1.2.1 Tracker Choice

The target tracker needs to provide track trajectory state estimation and the track quality measure. The probability of target existence is chosen for the track quality measure, mainly because the Melbourne University group has extensive and positive experience using trackers based on it.

Linear Multi-target Integrated Probabilistic Data Association (LM IPDA) target tracking filter has been chosen for track update. It has been shown that, in similar environment, this tracker is capable of tracking a large number of targets using reasonable computational resources. In a recent comparison carried under this evaluation, it is shown that even applying a particle filter based target tracker will not significantly improve the performance. Simple thresholding of the probability of target existence is to be used for target confirmation and termination decisions, and the thresholds are to be determined experimentally.

Automatic track initiation is to be applied. The system will assume no prior information on the number and position of the targets. However, to prevent from the system constantly initializing new tracks, only measurements obtained by sensor in the Surveillance Mode are to be used to initialize new tracks. Existing tracks are updated using measurements obtained by sensor in the Surveillance Mode, as well as measurements obtained by the sensor in the Track Update Mode.

1.2.2 Sensor Scheduling

Two kinds of sensor scheduling logic are to be employed. The baseline logic will use the round robin updates of each track in the Track Update sensor mode. The one step look ahead logic will choose which track to update based on the probabilities of target existence. The following goals of the optimization are identified: (i) minimize the number of tentative tracks*, i.e.,* increase the speed of false track discrimination. Each tentative track should either be confirmed or terminated as soon as possible; and (ii) minimize the time delay recognizing that the target has disappeared, or is lost; *i.e.,* quickly terminate confirmed tracks which are or have become false.

To achieve these aims, the following cost functions have been identified for tentative track i at time k with current value of the probability of target existence $\psi_k(i)$:

- Expected number of track updates for track i to get confirmed, given that it is a true track, and given current value of the probability of target detection, denoted here by $\tau_k^c(i)$; and
- Expected number of track updates for track i to get terminated, given that it is a false track, and given current value of the probability of target detection, denoted here by $\tau_k^t(i)$.

The total cost function of tentative track *i* at time *k* is $c_k(i) = \min\left(\tau_k^c(i), \tau_k^t(i)\right)$.

The following cost function has been identified for confirmed track *i* at time *k* with current value of the probability of target existence, $\psi_k(i)$:

- Expected number of track updates for track *i* to get terminated, given that it has become a false track, and given current value of the probability of target detection, denoted here by $\tau_k^t(i)$.

Total cost function of confirmed track *i* at time *k* is $c_k(i) = w_c \tau_k^t(i)$, where the weight factor w_c is used to determine the relative weighting between confirmed track update and tentative track update. In the Track Update Mode the costs for each track are determined before each dwell, and the track with minimal cost is updated.

2.3 Progress to Date

To date, the following progress has been achieved:

- Tracker choice of LM IPDA has been taken, based on tracking of large number of targets in significant clutter [Mus 2005]. Particle filter implementation has been shown to improve upon LM IPDA; however, the difference is not significant enough to warrant additional complexity and computational requirements [MusMore 2006].
- Sensor management scenario has been updated.
- Cost functions have been determined.

2.4 Future Work

Cost functions need to be evaluated, *i.e*. the functional descriptions of $\tau_k^c(i)$ and $\tau_k^t(i)$ have to be obtained and verified using simplified simulations. Proof of concept has to be achieved by integrated simulation and evaluation based on the scenarios and cost functions described. The following statistics are of importance:

- Average number of scans needed to confirm a true track;
- Average number of targets not followed by a true track;
- Average number of scans during which a confirmed false track is visible;
- Average life of confirmed false track;
- Maximum number of targets that can be updated before sensor resources run out.

These statistics have to be gathered for both baseline sensor operation and scheduled sensor operation. Some simplifications may need to be removed to obtain more realistic comparison between the base scenario, and cost based scenario. However, removing all the simplifications is not realistic in the time-frame/human resources available.

2 Sensor Scheduling for Passive Geolocation

2.1 Introduction

Sensor Scheduling for Passive Emitter Geolocation Evaluation aims to use sensors located on the networked UAVs. In [Okello] the problem of emitter geolocation using a set of netted UAVs was investigated. It achieves geolocation by processing measurements corresponding to a single pulse that is visible to each of the UAVs. Based on this method a minimum of three UAVs are required when the transmitter is known to be on the surface of the earth and four if the altitude of the transmitter is not known.

While this concept is quite easy to demonstrate, there are a number of problems that require solving before the method can be implemented in practice. These problems include: (i) the need to determine UAV trajectories for optimal geolocation; (ii) the need for an association algorithm capable of grouping together measurements from a common pulse given the shear number of pulses that are detected by each UAV; and (iii) the need for a suitable low bandwidth communication network for measurement transfer, again given the large number of pulses detected by each UAV.

Thus, given the bandwidth constraints, only a small number of measurements may be transferred between the UAVs, and an event when all UAVs involved actually receive the same pulse needs to be identified. The complexity of each of these problems is prohibitive when three or more netted UAVs have to be deployed.

2.2 Approach

In order to reduce the complexity of these problems, we consider a simpler system consisting of two netted UAVs in which geolocation is achieved by processing multiple time difference of arrival (TDOA) measurements collected over time by the UAVs as they traverse the surveillance region. Let us assume that the two UAVs move along a circle with a 180^o degree separation. When a pulse is detected by the two UAVs, a range difference r_{21} is generated and a processing algorithm then converts it to an appropriate set of 2-D coordinates that lie on a hyperbola. Figure 1 shows a typical scenario with one scanning radar emitter and two UAVs that move diametrically along a circle and includes the one–σ hyperbolae of range difference measurements. Over time the measurement hyperbola rotates and changes shape but remains pivoted at the emitter location.

A target tracking algorithm can be initialized and updated using these measurements, with the goal of estimating the emitter position. The problem of doing so is the extreme non–linearity of these measurements. In principle, a particle filter solution can be applied, but would require a huge number of particles to cover the uncertainty area. Instead we propose the following approach. The measurement uncertainty pdf depicted in Figure 25 can be approximated by a sum of Gaussian pdfs. Each element of the sum is a weighed converted measurement and starts one track component. Subsequent measurements can also be modeled by a sum of Gaussian pdfs, and used to update the track.

The proposed algorithm is a variant of the Integrated Track Splitting filter (ITS), proposed in [MusEvans 2006]. The track consists of components, one component for each history of the converted measurements. ITS recursively calculates the relative probability of these components. As the UAVs move and provide new measurements, the filter components which are far from the actual emitter position are going to diminish and be removed. Remaining components will converge on the radar emitter position with more accuracy over time.

For a stationary emitter, measures of performance will include speed of geolocation and accuracy of the location estimates. Constraints for the tracking problem include communication bandwidth, UAV trajectories, and complexity of association algorithm. The sensor scheduling problem can be described in this way: Given UAV speed, what is the optimum radius of the circle they traverse? The bigger the distance, the smaller distance between the one– σ hyperbolae and potentially more precise

measurements may be obtained. On the other hand, given the UAV speed, the bigger the distance the longer it will take them to rotate a given angle and provide hyperbolae intersection with small cross section.

We seek to find this optimum value, as well as determine the minimum time necessary to locate the emitter with a certain error. In the proof of concept phase, it is assumed that both UAVs and the emitter is in one plane. Adding the altitude of the UAVs and the curvature of the earth will complicate mathematics, while not impeding the proof of concept.

3.3 Progress to Date

The Gaussian Sum Measurement model and ITS tracker have been simulated and proven using highly nonlinear measurements.

3.4 Future Work

The first step is to find reasonable Gaussian sum presentation of the uncertainty shown in Figure 25. This involves determining the covariance and relative weight of each Gaussian sum element. The next step is to implement this scenario and simulate it for the proof of concept. Using the results, the optimum distance between UAVs, given speed and a priori distribution of the emitter should be determined.

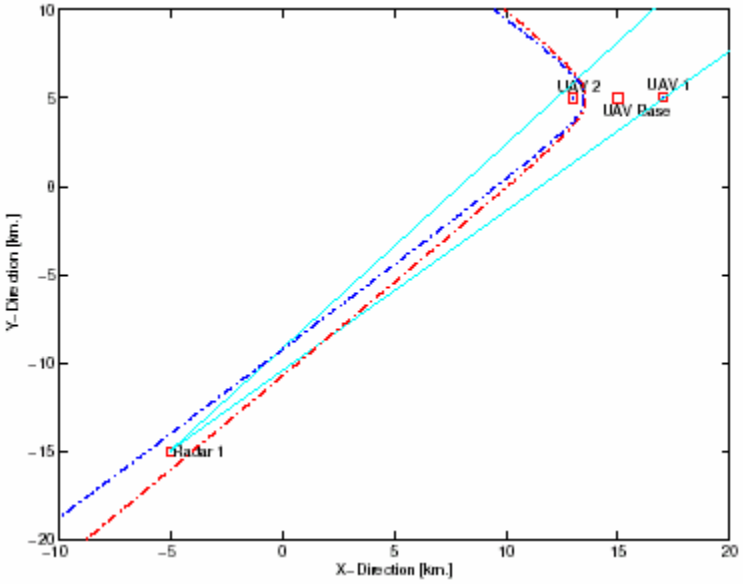

Figure 25: Scenario showing a scanning radar emitter, a pair of netted UAVs and the one−σ hyperbolae of the range difference measurements

2.A.6. FMAH Technical Progress

Overview

The goal of this portion of the project is to apply and adapt geometric diffusion methods of Coifman *et al.* to IR video data. As a simplification, we consider direct application of diffusion operators to the data itself.

Basic algorithm

Consider an image **u** consisting of a rectangular array of pixels, or more generally a rectangular array of vectors of fixed length (such an array of vectors might be, for

example, a gradient map or an optical flow field; we understand *pixel* to mean either scalar or vector values). We construct a diffusion filter **K**, such that repeated application of **K** to **u** suppresses the background while enhancing or preserving regions of interest in the image. Furthermore, we *reinitialize* the filter after a fixed number of steps, using the output of the previous iteration to generate a new filter which is then applied iteratively to the output of the previous iteration.

Construction of **K** proceeds as follows. For each pixel j in **u**, a group of neighboring pixels of size $(2n+1) \times (2n+1)$, denoted $\mathbf{x}(j)$ and center on the pixel j, is selected. Next, a scalar non-negative kernel function G is selected to measure the degree of similarity or difference of two groups $\mathbf{x}(i)$ and $\mathbf{x}(j)$. The function G is therefore a bivariate function of vectors; each vector is composed of groups of pixels. If $\mathbf{x}(i) = \mathbf{x}(j)$, then $G(\mathbf{x}(i),\mathbf{x}(j)) = 0$, and if $\mathbf{x}(i)$ and $\mathbf{x}(j)$ are "dissimilar" then $G(\mathbf{x}(i),\mathbf{x}(j))$ should be large. Usually, G is designed to be a symmetric function of its arguments, so that the degree of similarity between $\mathbf{x}(i)$ and $\mathbf{x}(j)$ is the same as the similarity between $\mathbf{x}(j)$ and $\mathbf{x}(i)$.

From this kernel function, together with an integer N and a small parameter ε, a matrix $\mathbf{K}'$ is constructed so that

$$\mathbf{K}'_{ij} = \begin{cases} e^{-G(\mathbf{x}(i),\,\mathbf{x}(j))/\varepsilon} & \text{if } |i-j| \le R \\ 0 & \text{otherwise} \end{cases}.$$

Then, **K** is constructed from $\mathbf{K}'$ by normalizing the row sums to one; that is, $\mathbf{K}_{ij} = \mathbf{K}'_{ij} / \sum_l \mathbf{K}'_{il}$ where the sum is over the columns. The matrix **K** may be viewed alternatively as a *diffusion operator*, since its action on **u** is identical to a forward time step in a discrete diffusion problem, or as a *Markov matrix*, since the sums of its rows are unity. Each row has $(2R+1)^2$ non-zero entries, and the total number of rows is equal to the number of pixels in **u**.

A relatively simple but powerful choice of G is given by the Euclidean distance $G(\mathrm{x}(i),\mathrm{x}(j)) = \|\mathrm{x}(i) - \mathrm{x}(j)\|^2$. In practice we use this choice exclusively, and instead apply transformations to **u** itself before computing the diffusion filter. We detail several examples of such transformations below.

Enhancements to the algorithm

As demonstrated in previous reports, the above algorithm is remarkably powerful for denoising/segmenting noisy IR images. Now we present a number of enhancements of the algorithm.

Namely, the filter **K** may be constructed from a transformed version of the input image **u**, denoted **f(u)**. For example, we might compute the gradient vector field of the image, then construct the filter from the gradient vector field. **K** could then be applied back on the image itself, or to the gradient field.

For the time being, we consider two transformations:

1. Local covariance matrix. In this case, consider a square patch of pixels centered on pixel i in the original image, as well as square patches of pixels center on neighboring pixels, for a total of (for example) 9 square patches of pixels. From this set of 9 vectors

we construct a 9 x 9 covariance matrix. This 9 x 9 covariance matrix is the vector-valued pixel *i* in the image **f(u)**. More generally, fewer or more vectors may be chosen to form the covariance matrix; a special case would be only 1 vector, in which case **f(u)** is an image whose pixels are local variances of **u**.

2. Two-scale variances. Consider two square patches of pixels, each of a different size, centered on pixel *i* in the original image. Now we compute the variance of each patch to form a vector of length 2, which is the vector-valued pixel *i* in the image **f(u)**.

Both of these transformations are meant to capture *local statistics* of the image, so that neighborhoods with similar local statistics will diffuse to one another in the diffusion process.

Thus, the segmentation algorithm has the following steps:

Let $\mathbf{u_1}=\mathbf{u}$ or **f(u)**. For m=1,…,M:

1. Define $\mathbf{K}_m$ using input $\mathbf{u_m}$ as above.
2. Compute $\mathbf{u_{m+1}}=(\mathbf{K}_m)^N\ \mathbf{u_m}$

The input to the algorithm is the original image **u.** The parameters of the algorithm are

- f, G,ε (and any parameters needed by f and/or G).
- R,n. The radius of the filter window and the radius of the pixel sets used to form local neighborhoods in the image.
- M,N. The number of outer and inner iterations segmentation iteration.

Application to moving target segmentation; optical flow

In many applications of interest, the goal is to separate or segment moving targets from background clutter. In such situations, the image is first processed using an *optical flow algorithm.* Such algorithms attempt to determine the local motion vectors of features within a sequence of image frames. The use of optical flow in conjunction with diffusion map processing was proposed by Mr. John Costello (Raytheon) and is an excellent example of the highly collaborative nature of our ISP II program.

Though a variety of optical flow algorithms are in use today, we highlight the use of one in particular[1], based on phase gradients.

Below, we apply the algorithm to a sequence of video frames shown in Figure 26. (The imagery chosen is a publicy-available visible-light video sequence; substantial testing on uncooled IR imagery was also done, but the results are not cleared for inclusion in this report). The goal of the algorithm is to segment the image into "Target" and "Nontarget" regions. After the segmentation process, the segmented image could be used to pull out the target portion of the image for further analysis (such as classification/identification, etc.); for now, we focus only on the segmentation process.

Figure 28 shows first frame of the sequence, together with the magnitude of the optical flow vectors. Since the camera is stationary, the background is relatively homogeneous, but some amount of noise is present.

[1] "Computation of component image velocity from local phase information." D.J. Fleet and A.D. Jepson, International Journal of Computer Vision, 5:1, 77-104 (1990).

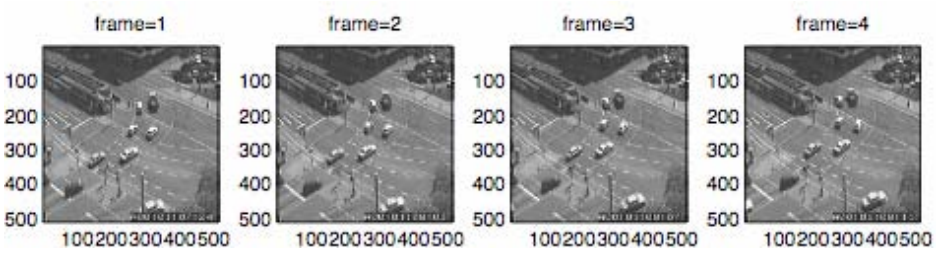

Figure 26: Four frames from the "Taxi" video sequence, courtesy Karlsruhe University Institute for Algorithms and Cognitive systems

Figure 29 shows the result of segmentation mapped back to the original frame. We apply thresholding to the segmented flow image, then use the result as a mask on the original image. Fairly good segmentation results are obtained, though some smaller moving objects such as pedestrians are missed, and the vehicle boundaries are not very well resolved.

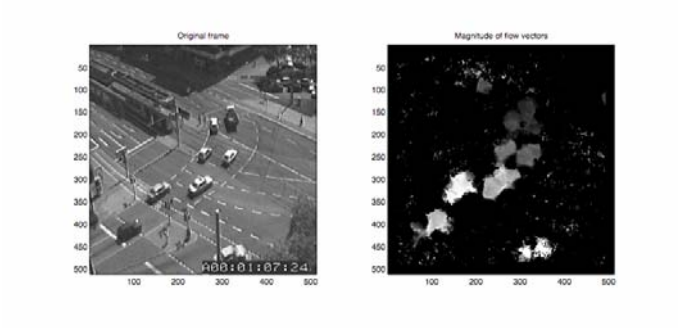

Figure 27: First frame of video sequence, together with the magnitude of the flow vectors computed via phase-gradient-based optical flow. Segmentation is already relatively good, though there is some noise present

Figure 29 shows the magnitude of the optical flow vectors before and after segmentation. In this case we used a special case of the local covariance matrix for **f**; **f(u)** is simply an image of local variances of **u.**

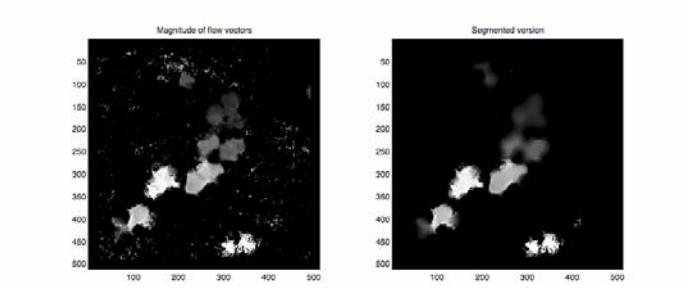

Figure 28: R=n=1, ε = 0.0003, M = 5, N = 30, f = local variances of patches of size 17x17. Segmentation consists mostly of removing small noisy regions

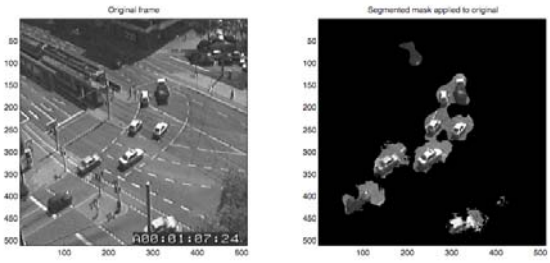

Figure 29: Pointwise product of original image with thresholded flow vector magnitude image. Parameters were tuned for vehicle-sized objects, so pedestrians were lost

The overall process applied to achieve the results of Figure 28 is:

Optical flow → Diffusion-based segmentation → Thresholding → Masking

Next, we consider the same sequence of images with an artificial panning motion. This is an attempt to simulate a video sequence in which the camera is situated on a moving platform. The camera pan is simulated by shifting the image to the left by a fixed amount in each successive frame. Figure 29 shows the original frame together with the optical flow magnitudes. Additional noise and background clutter is now present in the optical flow image. Figures 30 and 31 show the segmented and masked images. Results are slightly less clear than in the stationary case, but still quite good.

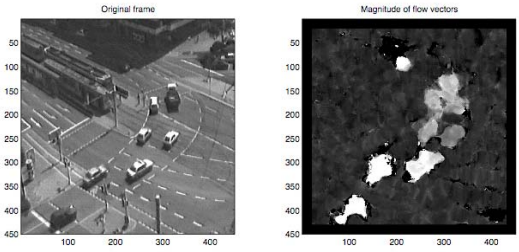

Figure 30: Optical flow applied to image sequence with camera pan. The background regions now contain noise due to the camera panning motion

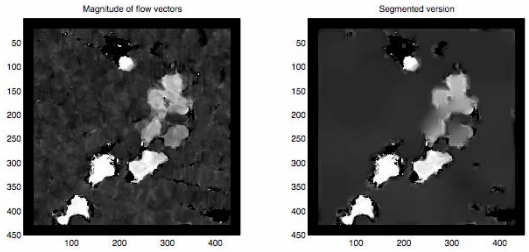

Figure 31: Segmented optical flow (R=n=1, ε=0.000002, M=10, N=100, f = local variances of patches of size 21 x 21. After diffusion the background is relatively homogeneous and much noise has been removed

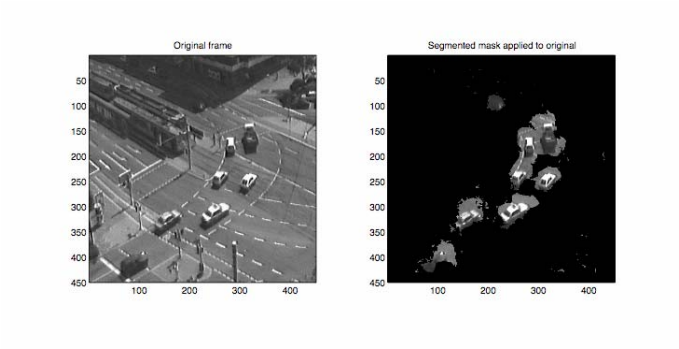

Figure 32: Masked frame after thresholding. Targets are fairly well segmented though some noise is still present

Note: FMAH also processed an extremely challenging set of UCIR imagery on-site at Raytheon. The results were extremely promising; however, the imagery is ITAR controlled and program-sensitive. As with the radar processing results discussed in Section 2.a.1.c, the results are available to the DARPA PM and the ONR COTR.

Application to SAR images

SAR imagery typically contains noise plus target with relatively good separation between the local statistics of the two. Therefore, we applied the diffusion filter based on multiscale variances to some sample SAR imagery. Though we cannot show the results in this report, the results were encouraging.

Possible next steps

- Test performance of algorithm on images at a variety of ranges, with the goal of developing a 'training' set of parameters for a given target at close range, which can then be used for target segmentation/identification at longer ranges.
- Develop algorithms for automated or semi-automated choice of parameters (currently parameters are selected through manual trial-and-error).
- Research optical flow algorithms and develop a better integration of optical flow and diffusion filters into a single unified algorithm for segmentation of images with moving targets.

2. B. Publications

There were no refereed publications that occurred during the current PoP.

1. Craig O. Savage and Bill Moran, "Waveform Selection For Maneuvering Targets Within An IMM Framework," IEEE Trans AES, accepted for publication.
2. A. Chhetri, D. Morrell and A. Papandreou-Suppappola, ``Non-myopic sensor scheduling and its efficient implementation for target tracking applications," EURASIP Journal on Applied Signal Processing, to appear 2006.
3. A. Chhetri, D. Morrell and A. Papandreou-Suppappola, ``On the use of binary programming for sensor scheduling," IEEE Transactions on Signal Processing, submitted February 2006.

2. C. Conference Proceedings

1. C. O. Savage, R. Cramer, and H. A. Schmitt, "Geolocation with the Unscented Kalman Filter," in Special Session on Cooperative Dynamic Systems, 2006 IEEE

International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, April 24, 2006.

2. N. N. Shah, A. Ramirez, D. Waagen, H. A. Schmitt and A. Hero, "Comparison of Inter-class Divergence for Linear and Nonlinear Dimensionality Reduction, with and without Class Labels," *Combat Identification Systems Conference (June 19-22, Orlando, Florida)* 2006, submitted.

3. N. N. Shah, B. Corner, D. Waagen, and H. A. Schmitt, "Correlation of Inter-class Divergence and Classification Performance," *Combat Identification Systems Conference (June 19-22, Orlando, Florida)* 2006, submitted.

4. R. Cramer, S. Bellofiore, T. Stevens, H. A. Schmitt, D. Waagen, N. Patwari, A. O. Hero, "Localization, Detection and Tracking for Wireless Sensor Networks," MSS, 2006, submitted.

5. N. Patwari and A. O. Hero III, "Indirect Radio Interferometric Localization via Pairwise Distance", Third Workshop on Embedded Networked Sensors (EmNets 2006) May 30-31, 2006, Cambridge, MA, accepted.

2. D. Consultative and Advisor Functions

There were two consultative or advisory functions that occurred during the current PoP. The first relates to a Raytheon Shooter Localization demonstration using the MICA-2/Z sensor nodes. This work is being funded under the DARPA IXO NEST Phase II program. The Phase I shooter localization algorithms were developed by VU. Preliminary results indicated that the shooter localization algorithm has significant potential. The program was subsequently classified and was ultimately transitioned to Raytheon for demonstration and refinement under Phase II. The DARPA IXO Program Manager has kindly given permission for several of these algorithms to be used in our ISP Phase II program. The Raytheon NEST program has identified a critical need for the development of an accurate sensor localization algorithm that is scalable to hundreds or thousands of nodes. Indeed, the DARPA NEST program hopes to demonstrate a 10,000 node network. We have identified and are evaluating several promising mathematical approaches to sensor localization developed by Al Hero (UM) and Bill Moran (UniMelb); these will be made available to the Raytheon NEST program if they are successful. Technical progress in these areas was discussed in the Thom Stevens and Sal Bellofiore support the DARPA ISP II and DARPA NEST programs, and, more generally, the two programs have developed a strong working relation.

The second function relates to optical flow test facility at Eglin, Air Force Base. Raytheon and Georgia Tech have had preliminary discussion with Dr. T.J. Klausutis of Eglin AFB about the possibility of using their facility to evaluate the Georgia Tech CADSP imager being investigated on our ISP Phase II program. While these discussions are preliminary, Dr. Klausutis was interested in learning more about the capabilities and maturity of the CADSP Imager, so a Technical Interchange Meeting (TIM) was held at Eglin and supported by Raytheon (Schmitt and Waagen) and Georgia Tech (Anderson). Dr. Klausutis has offered to make available a GPS-equipped truck capable of collecting ground-truth optical flow imagery should a "packaged" CADSP imager be ready in late fall 2006. Georgia Tech and Raytheon are working to take advantage of that opportunity.

2. E. New Discoveries, Inventions or Patent Disclosures
There were no patent disclosures filed during the current PoP.

2. F. Honors/Awards
There were no honors or awards received during the current PoP.

2. G. Transitions.
There were no technology transitions achieved during the current PoP. However, we plan to release the MATLAB simulation code for cooperative control of UAVs for passive geolocation to the US Air Force Academy.

2. H. References
[Patwari & Hero 2006] Neal Patwari and Alfred O. Hero III, "Indirect Radio Interferometric Localization via Pairwise Distance", Third Workshop on Embedded Networked Sensors (EmNets 2006) May 30-31, 2006, Cambridge, MA.

[Maroti *et al.* 2005] M. Maroti, B. Kusy, G. Balogh, P. Volgyesi, K. Molnar, A. Nadas, S. Dora, A. Ledeczi, "Radio Interferometric Positioning," *Technical Report, TR# ISIS-05-602*, Vanderbilt University, Nashville, Tennessee, 2005.

[Mellon *et al*. 2003] G. Mellen, II, M. Pachter and J. Raquet. "Closed-Form Solution for Determining Emitter Location Using Time Difference of Arrival Measurements", IEEE Trans. on Aerospace and Electronic Systems, 39(3):1056–1058, July 2003.

[Howard *et al.* 2004] S. D. Howard, S. Suvorova, and W. Moran, "Waveform libraries for radar tracking applications," International Conference on Waveform Diversity and Design, November 2004.

[Sera *et al.* 2006] S. P. Sira, A. Papandreou-Suppappola, D. Morrell, D. Cochran, "Waveform-Agile Sensing For Tracking Multiple Targets In Clutter," CISS, 2006.

[Van Trees] H. L. Van Trees, Detection Estimation and Modulation Theory, Part III. New York: Wiley, 1971.

[Kershaw & Evans 1997] D. J. Kershaw and R. J. Evans, "Waveform selective probabilistic data association," IEEE Trans. Aerospace and Electronic Systems., vol. 33, pp. 1180-1188, Oct. 1997.

[Foy 1976] W. Foy, "Position-Location Solutions by Taylor-Series Estimation," *IEEE Transactions on Aerospace and Electronic Systems,* Vol. 12, March, 1976.

[Julier *et al.* 1997] S. J. Julier, and J. K. Uhlmann, , "A New Extension of the Kalman Filter to Nonlinear Systems," in *SPIE International Symposium for Aerospace/Defense Sensing, Simuation and Controls,* April, 1997.

[Mus 2005] D. Musicki, "Limits of Linear Multitarget Tracking," 8th International Conference on Information Fusion, Philadelphia, Pa, July 2005.

[MusMore 2006] D. Musicki and M. R. Morelande, "Tracking a Large Number of Targets in Clutter with Particle Filter," accepted for 14th European Signal Processing Conference, Florence Italy, September 2006.

[Okello 2006] N. Okello, "Emitter Geolocation with Multiple UAVs" Accepted for 9th International Conference on Information Fusion, Florence, Italy, July 2006.

[MusEvans 2006] D. Musicki and R. J. Evans, "Measurement Gaussian Sum Mixture Target Tracking," accepted for 9th International Conference on Information Fusion, Florence, Italy, July 2006.

2. I. Acronyms

ADTS	Advanced Detection Technology Sensor
ASU	Arizona State University
ATA	Automatic Target Acquisition
AVU	Algorithms Verification Units
CADSP	Cooperative Analog Digital Signal Processor
CCDR	Classification Constrained Dimensionality Reduction
CRB	Cramér–Rao Bound
CROPS	Classification Reduction Optimal Policy Search
DARPA	Defense Advanced Research Projects Agency
DS	Danzig Selector
DSA	Distinct Sensing Area
dwMDS	Distributed, weighted, multi-dimensional scaling
FPA	Focal Plane Array
FMAH	Fast Mathematical Algorithms and Hardware
GEM	Geometric Entropy Maps
Georgia Tech	Georgia Institute of Technology
GPS	Global Positioning System
IASG	Independently Activated Sensor Group
ISP	Integrated Sensing and Processing
IXO	Information Exploitation Office
kNN	k-Nearest Neighbor
LEAN	Laplacian Eigenmap Adaptive Neighbor
LIP	Linear Integer Programming
M2M	Multipoint-to-multipoint
MC	Monte-Carlo
MTT	Multi-target tracking
NEST	Networked Embedded System Technology
NDA	Non-disclosure Agreement
NLIP	Nonlinear Integer Programming
NLOS	NetFires Non-Line of Sight
NUC	Non-Uniformity Compensation
ONR	Office of Naval Research
OSE	Out-of-sample extension
PAM	Precision Attack Munition
PDA	Probabilistic Data Association
PWF	Polarization Whitening Filter
PoP	Period of Performance
RIM	Radio Interferometric Measurements
RIPS	Radio Interferometric Positioning
RISCO	Raytheon International Support Company
RSS	Received Signal Strength
TAA	Technical Assistance Agreement

TDOA	Time Difference of Arrival
TIM	Technical Interchange Meeting
UAV	Unmanned Aerial Vehicle
UCIR	Uncooled infrared imaging
UKF	Unscented Kalman filter
UM	University of Michigan
UniMelb	Melbourne University
VM	Virtual Measurement
VU	Vanderbilt University